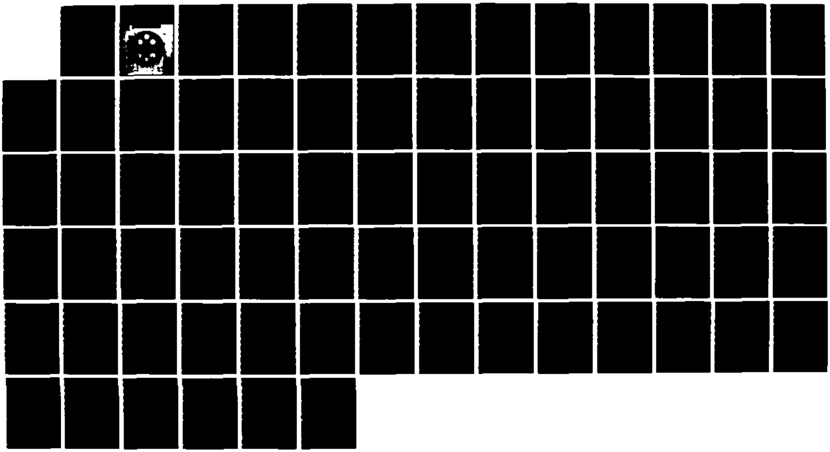
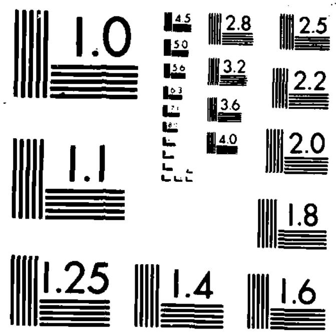
AD-A174 733 A COMPARISON OF STABILITY AND CONVERGENCE PROPERTIES OF TECHNIQUES FOR IN..(U) BROWN UNIV PROVIDENCE RI LEFSCHETZ CENTER FOR DYNAMICAL SYSTE.. 1/1

UNCLASSIFIED H T BANKS ET AL. JAN 86 LCDS-86-3 F/G 12/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(2)

A Comparison of Stability and Convergence Properties of Techniques for Inverse Problems

by

H.T. Banks and D.W. Iles

January 1986 LCDS #86-3

Lefschetz Center for Dynamical Systems

Division of Applied Mathematics Brown University Providence RI 02912

A Comparison of Stability and Convergence Properties of Techniques for Inverse Problems

by

H.T. Banks and D.W. Iles

January 1986 LCDS #86-3

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR- 86-2036	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Comparison of Stability and Convergence Properties of Techniques for Inverse Problems		5. TYPE OF REPORT & PERIOD COVERED Reprint 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) H.T. Banks and D.W. Iles		8. CONTRACT OR GRANT NUMBER(s) AFOSR 84-0398
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lefschetz Center for Dynamical Systems Division of Applied Mathematics Brown University, Providence, RI 02912		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2304/A1
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research Bolling Air Force Base Washington, DC 20332 nm		12. REPORT DATE January 1986 13. NUMBER OF PAGES 67
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) same as 11		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release: distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Approved for public release; distribution unlimited.

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Included

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A Comparison of Stability and

Convergence Properties of Techniques

for Inverse Problems*

H.T. Banks+ and D.W. Iles
Lefschetz Center for Dynamical Systems
Division of Applied Mathematics
Brown University
Providence, RI 02912

January 1986

* This research was supported in part by the National Science Foundation under NSF Grant MCS-8504316, the Air Force Office of Scientific Research under Contract AFOSR-84-0398, and the National Aeronautics and Space Administration under NASA Grant NAG-1-517.

+ Part of this research was carried out while the first author was a visiting scientist at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA, which is operated under NASA Contracts NAS1-17070 and NAS1-18107.

Abstract

We consider a series of numerical examples and compare several algorithms for estimation of coefficients in differential equation models. Unconstrained, constrained and Tikhonov regularization methods are tested for the behavior with regard to both convergence (of approximation methods for the states and parameters) and stability (continuity of the estimates obtained with respect to perturbations in the data or observed states).

The Table of Contents

Introduction

In [1], [2] Banks et al. describe an algorithm for solving a two dimensional version of the parameter identification problem for

$$(1) \qquad D(qDu) = f, \qquad x \in [0,1],$$

where q is an unknown to be chosen from some given parameter set Q, f is assumed known, and observations $\hat{u}$ are given for $u=u(q)$. Here $D = \partial/\partial x$ while in [1], [2] one replaces D with an appropriate 2-dimensional gradient operator. The algorithm given in [1], [2] is based on spline approximations u^N, $q^M \in Q^M$ for both the states u and unknown parameters q and a least squares criterion

$$(2) \qquad J(q) = \int_0^1 |\, u(q)-\hat{u}\,|^2 \, dx$$

A convergence theory (as the dimensions of the approximating spline spaces increase i.e., $N \to \infty$, $M \to \infty$) is given in [1] where one may use either linear or cubic splines for the state approximations and cubic splines for the parameter approximations. An essential feature of these particular convergence proofs is that the admissible parameter set Q and its approximations Q^M lie in some compact subset of C[0,1]. This same compactness assumption plays a fundamental role in proving stability (e.g., continuity of the inverse of the mapping from the parameter estimates to the observations or data) as is discussed in [3], for example. As we shall seek to demonstrate in this report, it so happens that this compactness is also important in computational aspects of the algorithms.

Perhaps the most direct way to interpret the compactness requirement is in terms of constraints on the parameters. For example, in the computations reported herein, we imposed compactness in C[0,1] by putting upper and lower bounds on the function values as well as an upper bound on the absolute values of the slope of the functions. The results presented in this paper illustrate the apparent necessity in many examples of including such a compactness constraint in computational algorithms. Examples are given below where both stability and convergence properties are as expected whenever a constrained estimation procedure is employed whereas instability and divergence is in evidence when unconstrained techniques are used. It is safe to speculate that similar behavior occurs in problems with parabolic and hyperbolic systems as well as elliptic. In our own work and in that reported in the literature - e.g. see Yoon and Yeh [7], one sometimes encounters severe problems with oscillations in the estimates for q as one pushes the algorithms for increased accuracy in the parameter estimates. As the examples in this report demonstrate, these difficulties can to some extent be alleviated by imposition of compactness constraints.

An alternative but essentially theoretically equivalent approach involves the use of Tikhonov regularization as formulated by Kravaris and Seinfeld in [5]. One restricts the parameter set to $Q_R \subset Q$ with Q_R compactly imbedded in Q and then modifies the original least squares criterion J to minimize $J_\beta = J + \beta \|q\|_R^2$ where $\|\cdot\|_R$ is the norm in Q_R and β is a regularization parameter. Thus minimizing sequences for J_β are bounded in Q_R and hence compact in Q; this is, in some sense, roughly equivalent to minimizing J over a restriction of Q which is compact even though the minimization of J_β only produces (hopefully) an approximation to the minimizer for the original criterion J. In the cases considered below, we use $Q_R = H^1$ while $Q = C$ (which corresponds to $\Lambda = C^1$ and $\mathbf{R} = H^2$ in the notation of [5]).

As we shall see below, each approach has inherent difficul- .s in choosing related imbedding parameters: in the first, the estimates produced are sensitive to the constraints (the bound L on the derivatives of the parameters in the computations summarized in this report) while the estimates produced using regularization are quite sensitive to the regularization parameter β.

In our calculations we have attempted to compare the spline based algorithms on a number of examples for three cases: the unconstrained minimization of J; the constrained minimization of J; and unconstrained minimization of a regularized criterion J_β.

We briefly outline the algorithms we have used, deferring some of the details to Appendices.

The Estimation Algorithms

The methods we illustrate here are based on linear spline approximations for the states u in (1) as well as for the parameter q. Full details of the approximation schemes are given elsewhere (e.g. see [1], [2] and some of the references contained therein). We consider (1) in weak form and use the approximate equations

$$\langle qDu, D\phi_i \rangle + \langle f, \phi_i \rangle = 0, \qquad i = 1, \ldots, N\text{-}1 \tag{3}$$

where ϕ_i are the N−1 spline basis elements

$$\phi_i(x) = \begin{cases} x - (i-1)/N, & x \epsilon\ [\,(i-1)/N,\ i/N\,) \\ (i+1)/N - x, & x \epsilon\ [\,i/N,\ (i+1)/N\,) \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The aim is to find an approximate estimate of q,

$$q^M = \sum_{j=1}^{M+1} \rho_j \psi_j \tag{5}$$

where the ψ_j are the restrictions to [0,1] of the basis elements

$$\psi_j(x) = \begin{cases} x - (j-2)/M, & x \epsilon\ [(j-2)/M,\ (j-1)/M\,) \\ j/M - x, & x \epsilon\ [\,(j-1)/M,\ j/M\,) \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

These spline basis elements are not normalised to have a maximum value of one. In the unnormalised form the slopes are plus and minus one, which makes it very easy to express constraints on the slope as the difference between the coefficients of successive basis elements, without any scaling factors involving N or M.

Given an estimate q^M and substituting it into (3) we can solve for the corresponding u^N, where

$$(7) \qquad u^N = \sum_{i=1}^{N-1} \sigma_i \phi_i$$

Thus u^N is a function of q^M, i.e. $u^N(q^M)$ (see Appendix B).

The estimation algorithms estimate q by optimizing over q^M in a specified subset of C[0,1] the functional

$$(8) \qquad J^N(q^M) = \int_0^1 |u^N(q^M)-\hat{u}|^2 dx$$

The Unconstrained Algorithm

In this algorithm the functional is taken to be :

$$(9) \qquad J^N(q^M) = \int_0^1 |u^N(q^M)-\hat{u}|^2 dx$$

This functional was optimized using the reduced gradient algorithm (Appendix A), without imposing any constraints.

The Constrained Algorithm

This algorithm is similar to the unconstrained algorithm, but the estimates, q^M, are constrained to a compact subset of C[0,1]. A suitable compact subset of C[0,1] is the set of bounded functions with bounded derivative almost everywhere, so the constraints imposed on q^M were $|Dq^M| \leq L$, and $0.5 \leq q^M \leq 10.0$. The functional (9) was minimized using the reduced gradient algorithm (Appendix A) with these constraints.

The bound on the absolute value of the derivative is the parameter L. The appropriate value of this parameter to be used must also be estimated, either from apriori knowledge about the parameter q, or by looking at the behavior of the estimates as the constraints are eased, i.e., L is increased.

The constraint that the function be bounded is never significant in the work presented in this paper; however as we shall see, the results are quite sensitive to the derivative bound L.

Tikhonov Regularization

Tikhonov regularization is one method which has been proposed [5] to prevent the oscillations in the estimates. The estimation problem is changed by optimizing over a different functional $J^N_T(q^M)$, where :

$$J^N_T(q^M) = J^N(q^M) + \beta \| q^M \|_a \tag{10}$$

and

$$J^N(q^M) = \int_0^1 | u^N(q^M) - \hat{u} |^2 dx \tag{11}$$

$$\| q^M \|_a = a \int_0^1 | q^M |^2 dx + \int_0^1 | Dq^M |^2 dx \tag{12}$$

This functional was mimimized using the reduced gradient algorithm (Appendix A) without imposing any constraints.

As we have indicated, the addition of $\beta \| q^M \|_a$ essentially constrains the estimates to a compact subset of the estimation space, but it also creates some bias in the estimate. In the limit as N, M $\to \infty$ one cannot expect to converge to the true parameter unless $\beta \to 0$ as N and M $\to \infty$

The choice of β and a is somewhat arbitary, similar to the problem of choosing L in the constrained algorithm. This problem is discussed in a later section.

"Data" For The Algorithm

Throughout this work we assume that we know the functions $\hat{u}$,q and f as continuous functions of x. That is, in the test examples, we are given (independent of the approximation indices N and M) values for $x \rightarrow \hat{u}(x)$ (though possibly with some error), not just a finite dimensional approximation to it. Thus the least squares functional (9) is evaluated using an infinite dimensional value for $\hat{u}$. We do not look at the effect of knowing only a finite dimensional approximation to $\hat{u}$, as would be the case if u were observed only for some values of x and then an interpolated function were used for the observations $\hat{u}$.

The Error Measure: L^2 vs L^∞

In some of the results presented in this paper we have looked at the L^∞ norm of the estimate error. We could also have used the L^2 norm of the estimate error.

It is possible to have convergence of an estimate in the L^2 norm but not in the L^∞ norm but not visa versa, of course. However in the examples studied here the L^2 norm of the error has behaved similarly to the L^∞ norm. The oscillations which have developed have not tended to occur over a shorter length as their height increases, thus the L^∞ and L^2 norms of the error have increased together.

On those figures which show the changes in estimate error with iteration number both the L^2 and L^∞ norms of the estimate errors are given.

Examples

The following examples were used in testing the algorithms. For examples 2 through 5, f tends to infinity near $x=0$, and thus the calculation (see Appendix C) of F_1 ($=\langle f,\phi_1\rangle$) using Simpson's rule cannot be expected to be accurate. Therefore, for these examples F_1 was evaluated using its analytic expression.

Example 1

(13) $$q = \begin{cases} 2+x & x\epsilon[0,1/3) \\ 8/3-x & x\epsilon[1/3,\ 1] \end{cases}$$

(14) $$u = \sin(\pi x)$$

Example 2

(15) $$q = \begin{cases} 2+x, & x\epsilon[0,1/3) \\ 8/3-x, & x\epsilon[1/3,2/3) \\ 4/3+x, & x\epsilon[2/3,1] \end{cases}$$

(16) $$u = \sqrt{x}\,(1-\sqrt{x})$$

Example 3

(17) $$q = \begin{cases} 2 + 2*x, & x\epsilon[0,1/3) \\ 10/3 - 2*x, & x\epsilon[1/3,2/3) \\ 2/3 + 2*x, & x\epsilon[2/3,1] \end{cases}$$

(18) $$u = \sqrt{x}\,(1-\sqrt{x})$$

Example 4

$$(19) \qquad q = 2, \qquad x \epsilon [0,1]$$

$$(20) \qquad u = \sqrt{x}\,(1 - \sqrt{x})$$

Example 5

$$(21) \qquad q = \begin{cases} 2+x, & x \epsilon [0,1/3) \\ -3*x^2 + 3*x + 5/3, & x \epsilon [1/3,2/3) \\ 4/3+x, & x \epsilon [2/3,1] \end{cases}$$

$$(22) \qquad u = \sqrt{x}\,(1 - \sqrt{x})$$

The initial guess for q^M

The initial guess for q(x) in all Examples was $q_0^M(x) \equiv 1$.

The Constrained versus Unconstrained Algorithms

In general constraints are necessary for the estimation algorithm to work. We first compare the two algorithms taking the constraint parameter L to be one, i.e. we require that the absolute value of the slope of the estimates be less than or equal to one.

For Example 1 both algorithms perform similarly for N and M of comparable magnitude, but as M is increased only the estimates given by the constrained algorithm are accurate. The estimate given by the unconstrained algorithm develops oscillations (Fig 1 – Fig 4).

These oscillations are reduced as N is increased, but for the algorithm to perform satisfactorily at low N (a case of importance in actual applications of the method to more complex problems) the constraints are essential.

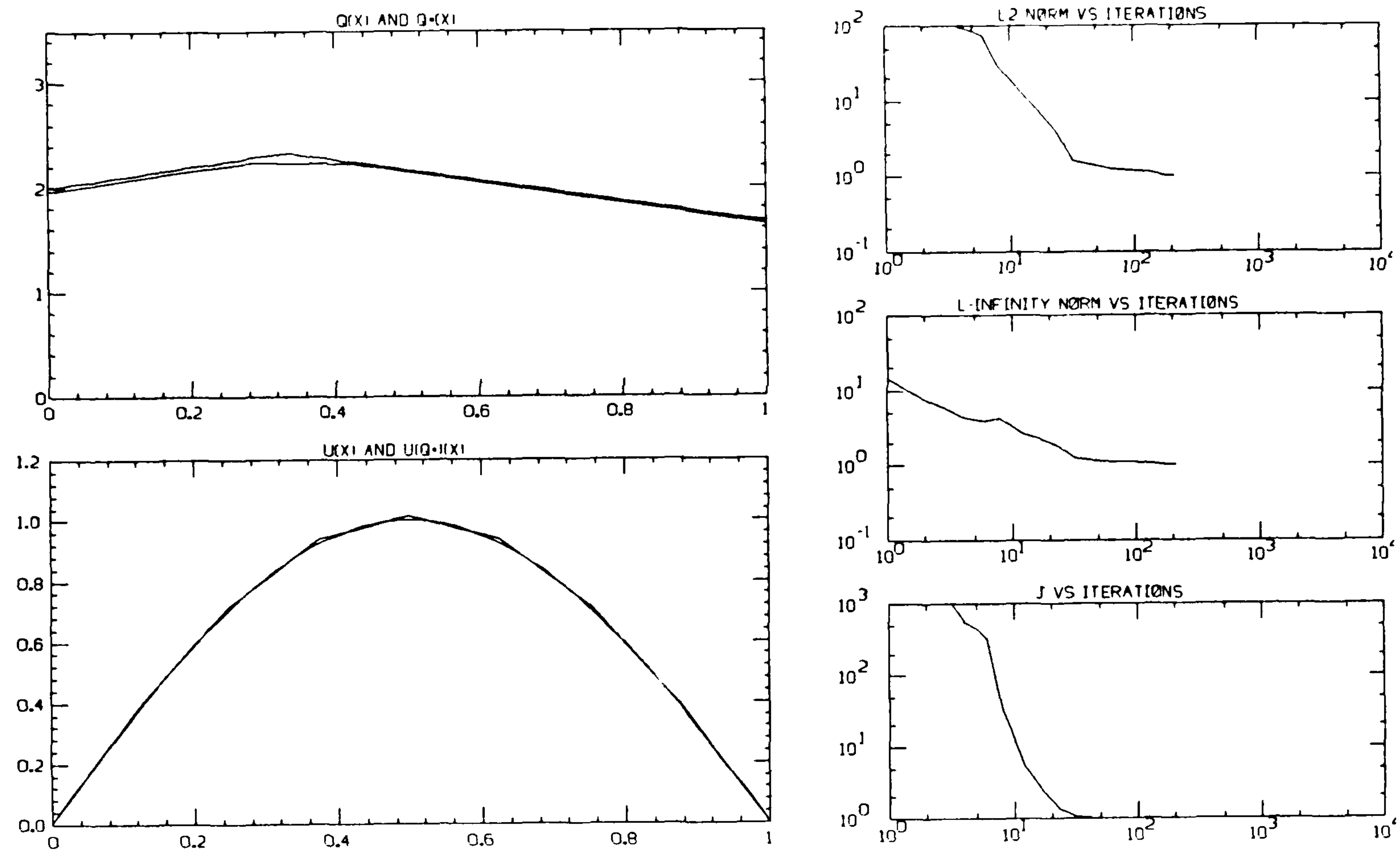
The situation is much more dramatic with u taken as $\sqrt{x}(1-\sqrt{x})$, where u is a much more sharply curving function near $x=0$ and hence much harder to approximate with u^N when N is small. For $N=8$ the constrained estimate shows little detail but does give an idea of the mean value of the true estimate (Fig 5). The unconstrained estimate is completely wild and gives no useful information (Fig 6).

That the unconstrained estimate is so bad is not the result of ill choosen convergence criteria. The best estimate, found using the knowledge of the true q is shown in Fig 7. The estimate using much weaker convergence criteria is shown in Fig 8. Neither give much information about the true q.

Figures 9 – 11 show the behavior of the unconstrained estimate as N is increased. For $N=96$ (Fig 11) the unconstrained estimate is good with only small oscillations at each end, and shows a reasonably steady improvement as J decreases, i.e. is insensitive to the convergence criteria used, compared to the estimate for $N=16$ $M=16$ (Fig 9) which deteriorates as you push the cost (J) to lower values (see the graphs on Figures 9 – 11 of L^∞ and L^2 vs iterations).

On the other hand the constrained algorithm behaves well for all the values of N (Fig 12 – Fig 14) and is insensitive to when you stop the iterations even for the $N=8$ $M=15$ case. (Fig 5). It also provides consistently better estimates with far fewer iterations even when $N=96$, although the number of iterations involved in finding the unconstrained estimate could be reduced considerably by using a better unconstrained optimization algorithm than the reduced gradient method. When used with no constraints the reduced gradient method is just a gradient method in a space isomorphic to the estimate space and will not, in general, perform any better than the normal gradient method.

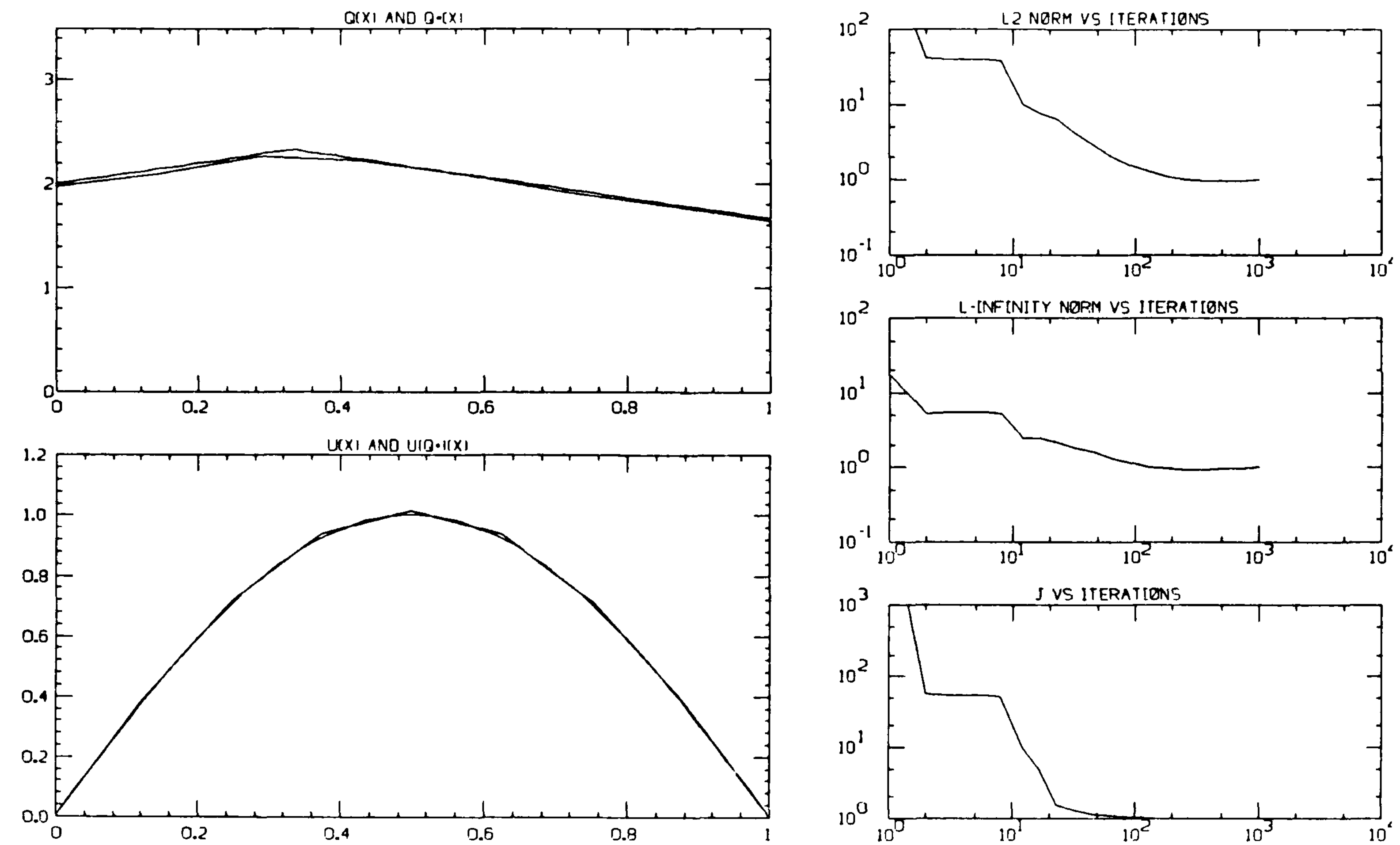
Fig 1: Constrained Estimation #1: N=8 M=7



ITERATIØN 209 J = 0.172774D-04

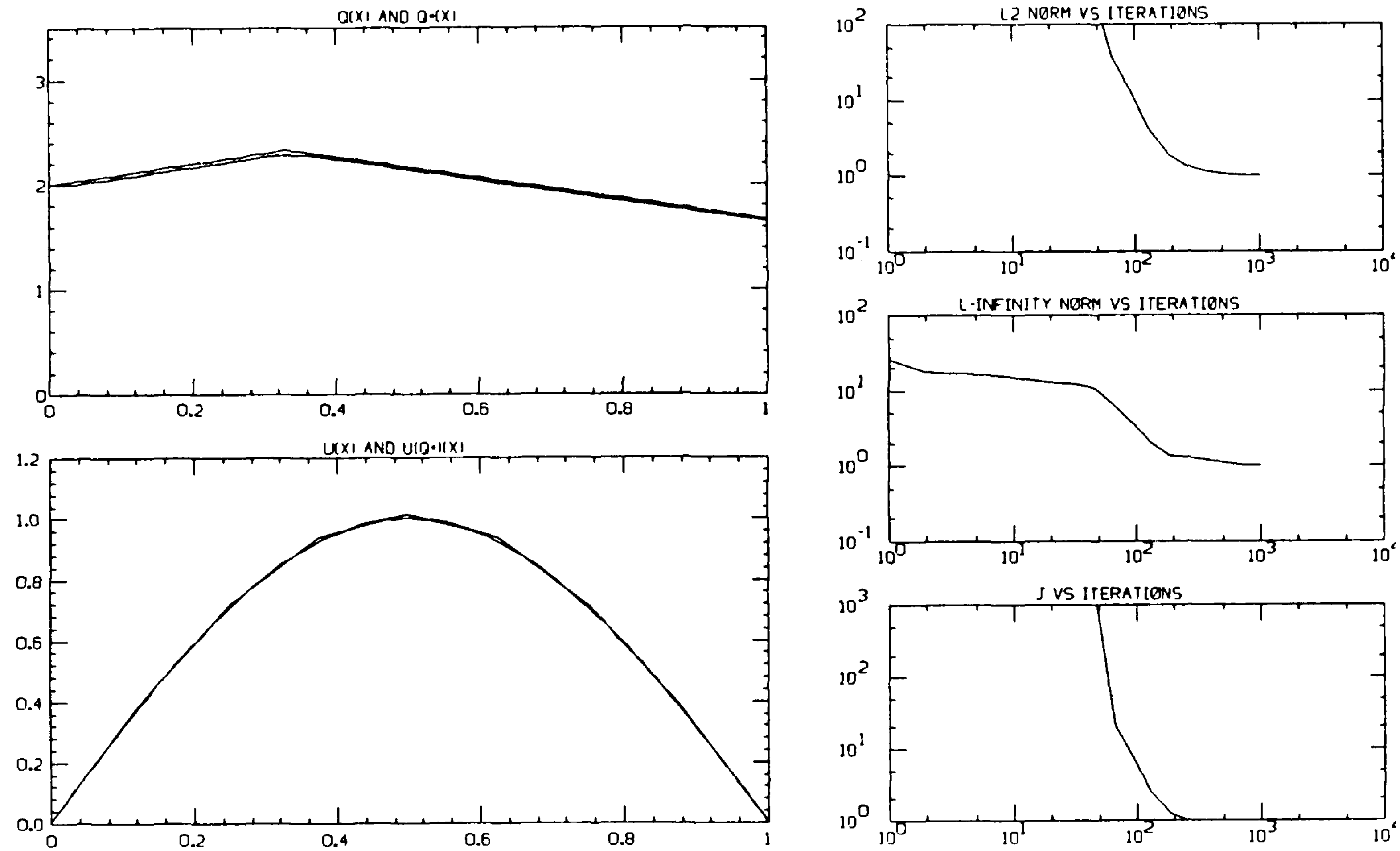
L2 NØRM Q*-Q = 0.121607D-02 L-INFINITY NØRM Q*-Q = 0.911284D-01

Fig 2: Unconstrained Estimation #1: N=8 M=7



ITERATIØN 999
J = 0.172357D-04
L2 NØRM Q*-Q = 0.103982D-02
L-INFINITY NØRM Q*-Q = 0.784441D-01

Fig 3: Constrained Estimation #1: N=8 M=50



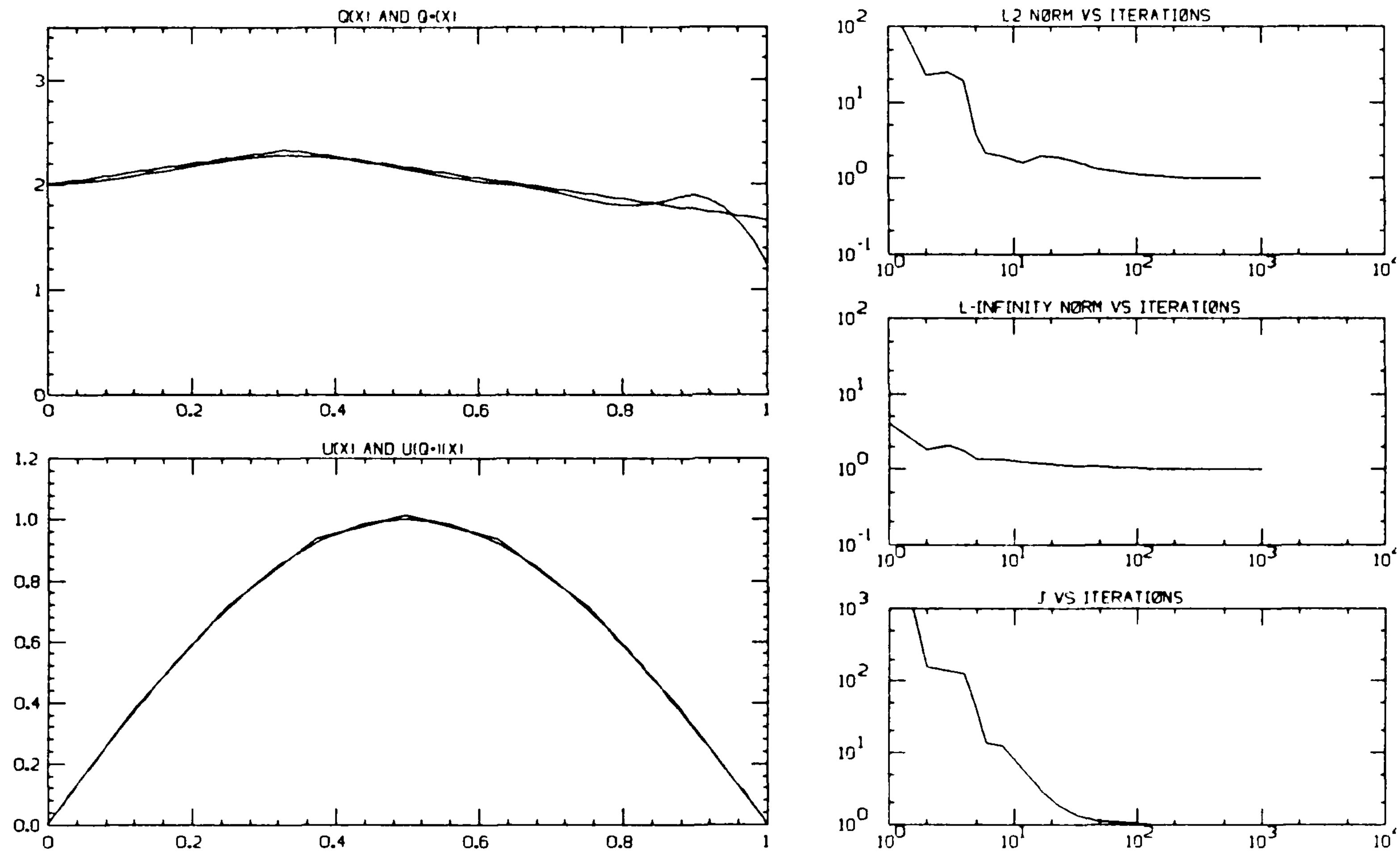
ITERATIØN 999

J = 0.171568D-04

L2 NØRM Q*-Q = 0.719631D-03

L-INFINITY NØRM Q*-Q = 0.502824D-01

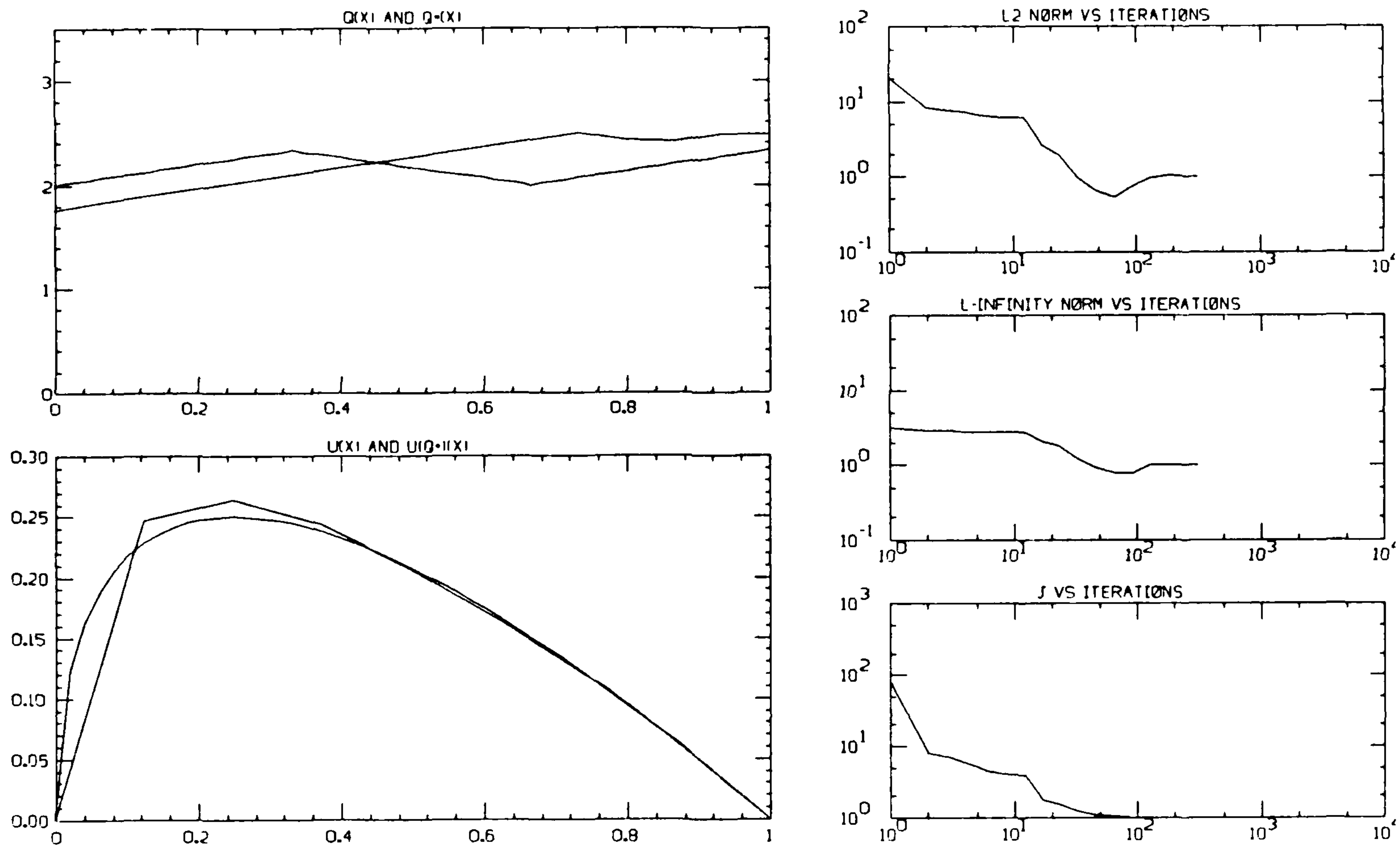
Fig 4: Unconstrained Estimation #1: N=8 M=50



ITERATIØN 999 J = 0.171505D-04

L2 NØRM Q*-Q = 0.427138D-02 L-INFINITY NØRM Q*-Q = 0.320491D+00

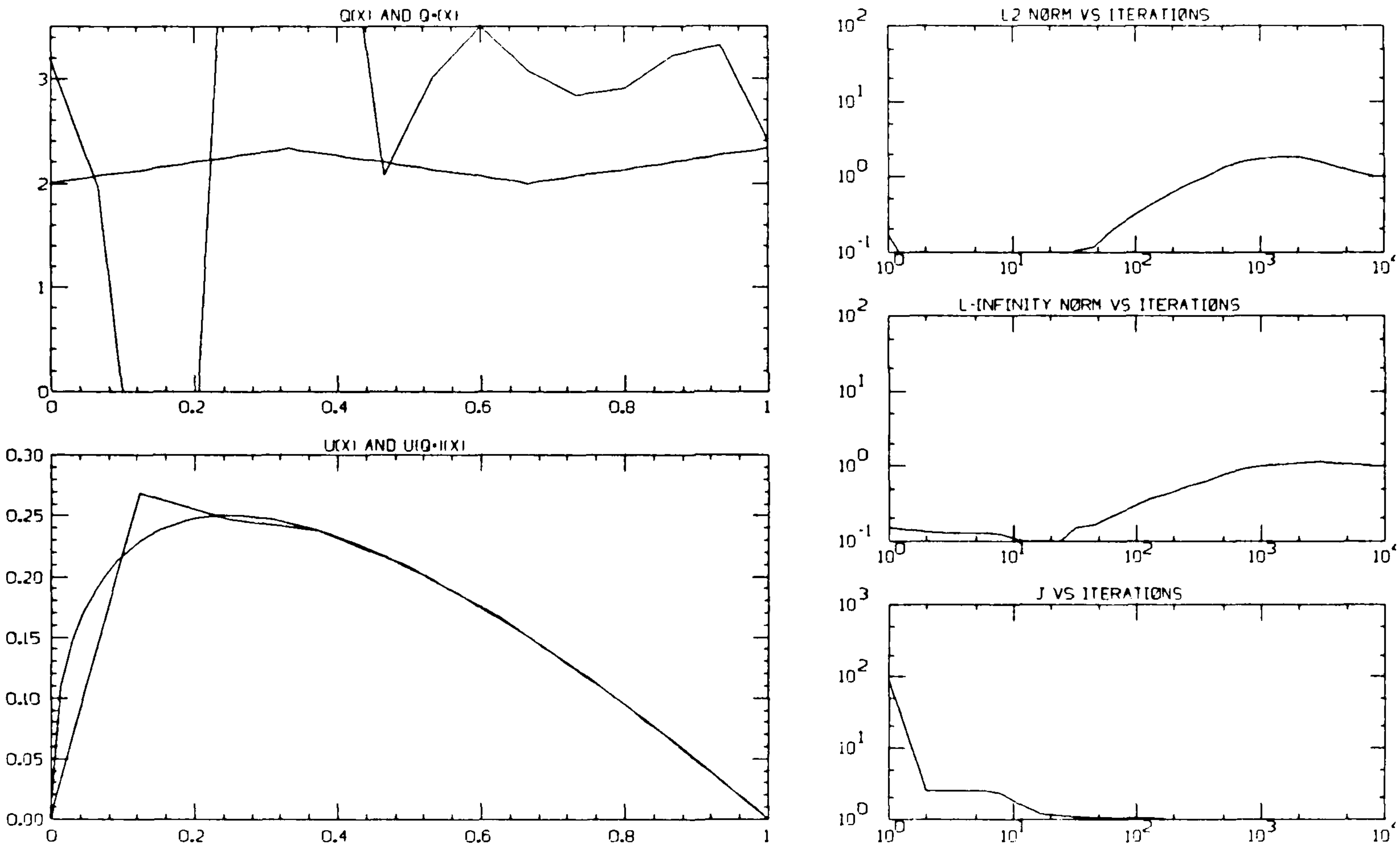
Fig 5: Constrained Estimation #2: N=8 M=15



ITERATIØN 306 J = 0.449928D-03

L2 NØRM Q*-Q = 0.654242D-01 L-INFINITY NØRM Q*-Q = 0.429525D+00

Fig 6: Unconstrained Estimation #2: N=8 M=15



ITERATIØN 9999 J = 0.395000D-03

L2 NØRM Q*-Q = 0.823974D+01 L-INFINITY NØRM Q*-Q = 0.875382D+01

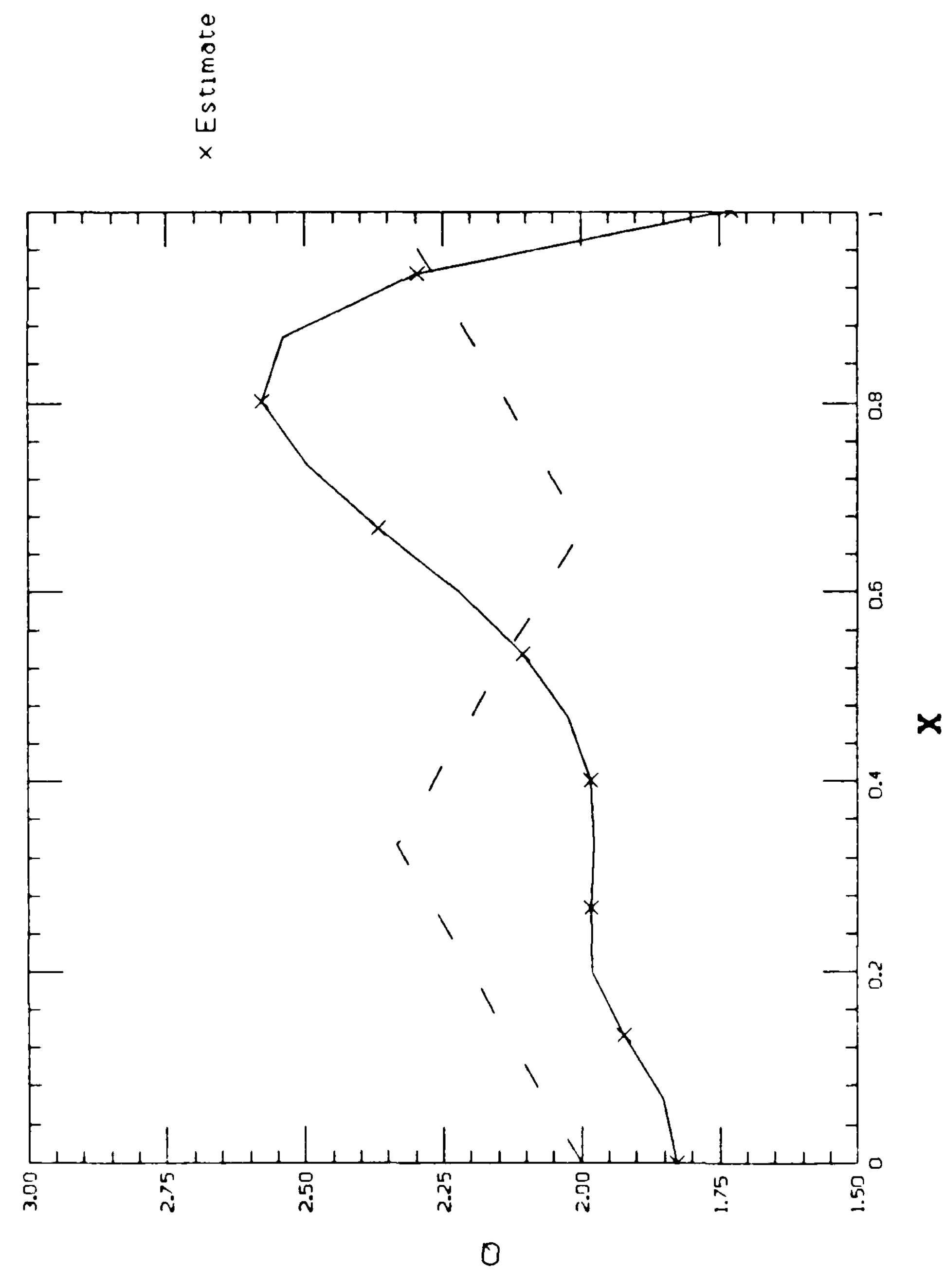
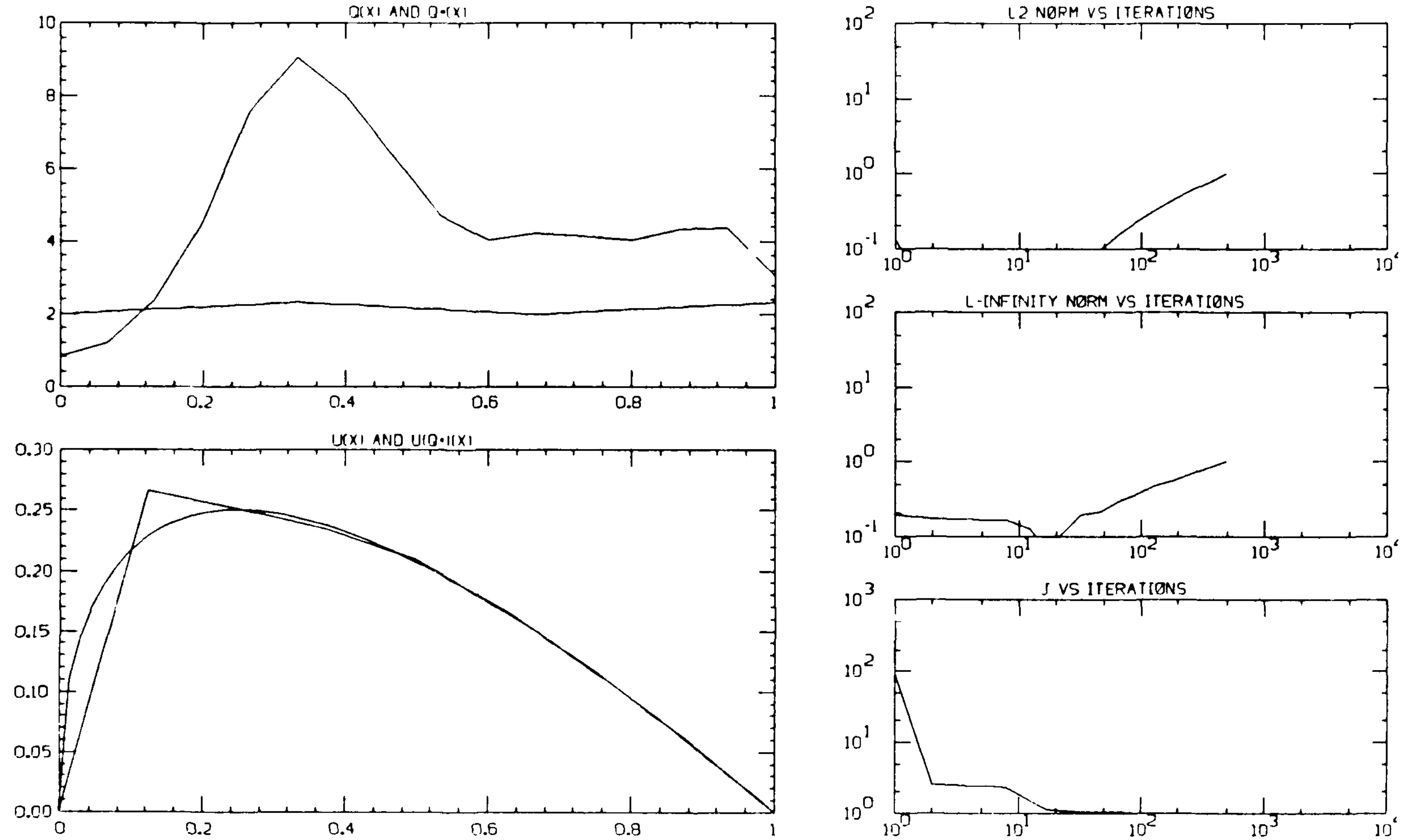


Fig 7: Best Estimate of Q (Uncon.) #2: N=8 M=15

Fig 8: Unconstrained Estimation (weak cvge) #2: N=8 M=15



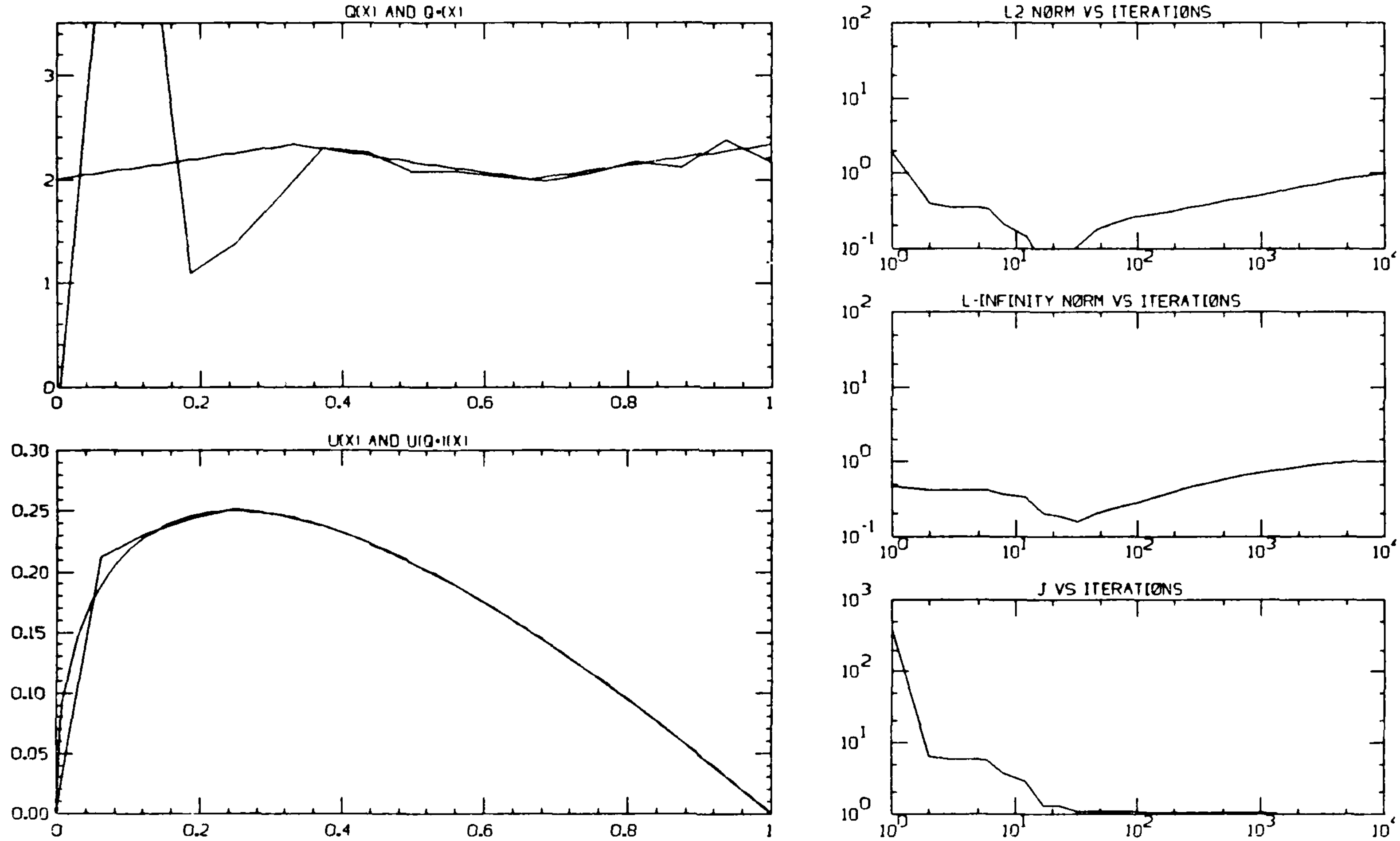
ITERATIØN 482

J = 0.399913D-03

L2 NØRM Q*-Q = 0.106111D+02

L-INFINITY NØRM Q*-Q = 0.670483D+01

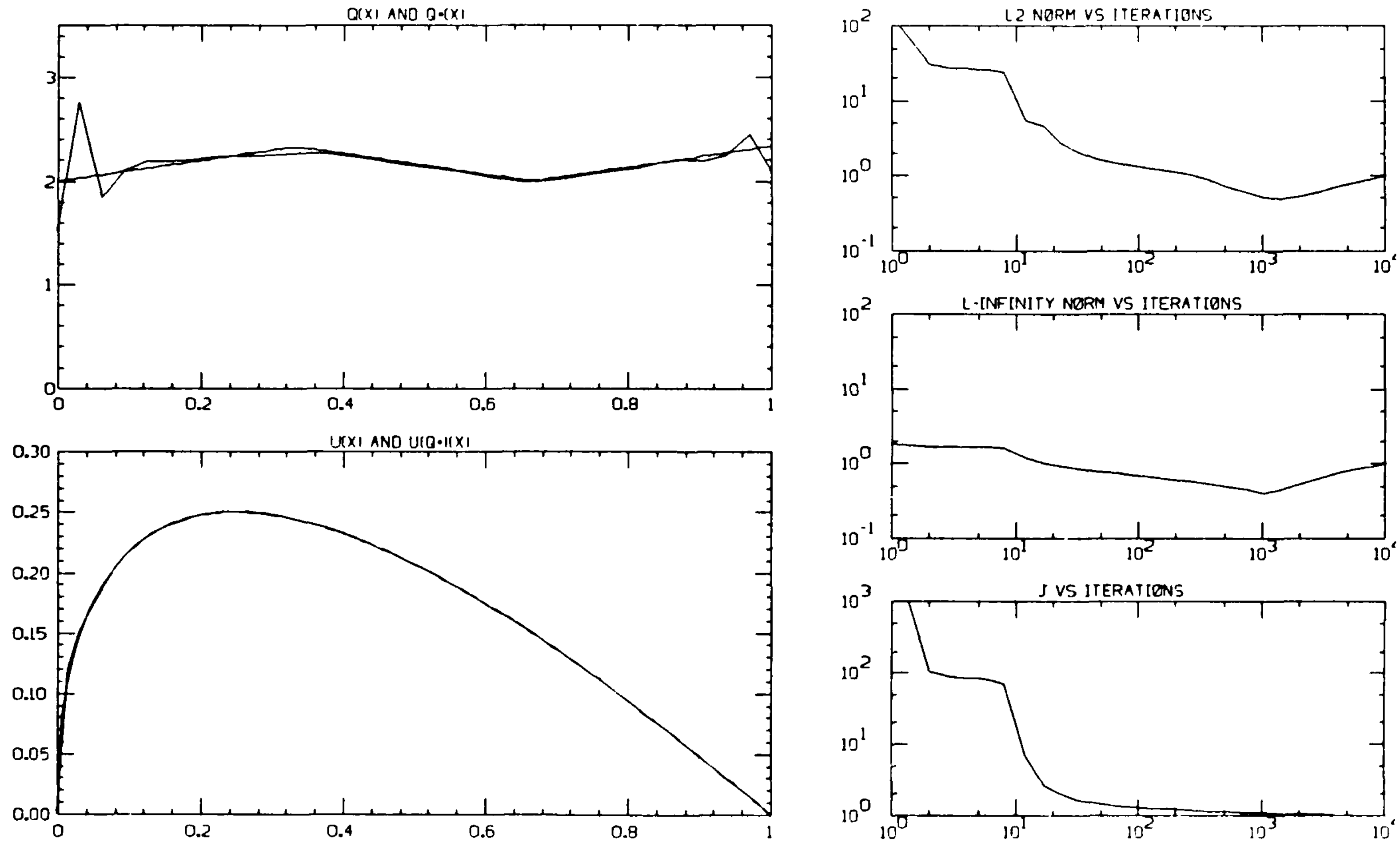
Fig 9: Unconstrained Estimation #2: N=16 M=16



ITERATIØN 9999 J = 0.100169D-03

L2 NØRM Q*-Q = 0.714982D+00 L-INFINITY NØRM Q*-Q = 0.279650D+01

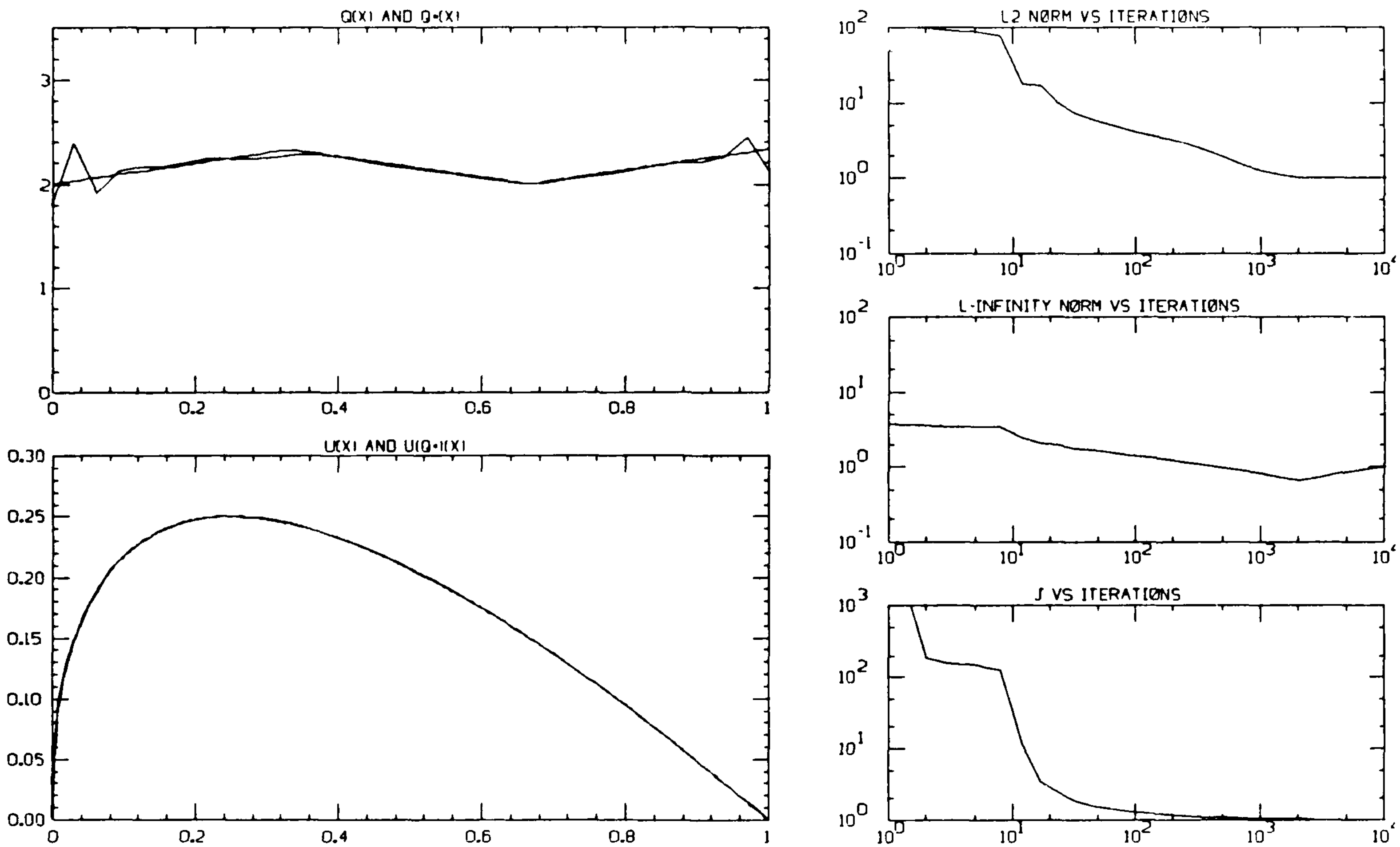
Fig 10: Unconstrained Estimation #2: N=64 M=32



ITERATIØN 9999 J = 0.541171D-05

L2 NØRM Q*-Q = 0.104115D-01 L-INFINITY NØRM Q*-Q = 0.732284D+00

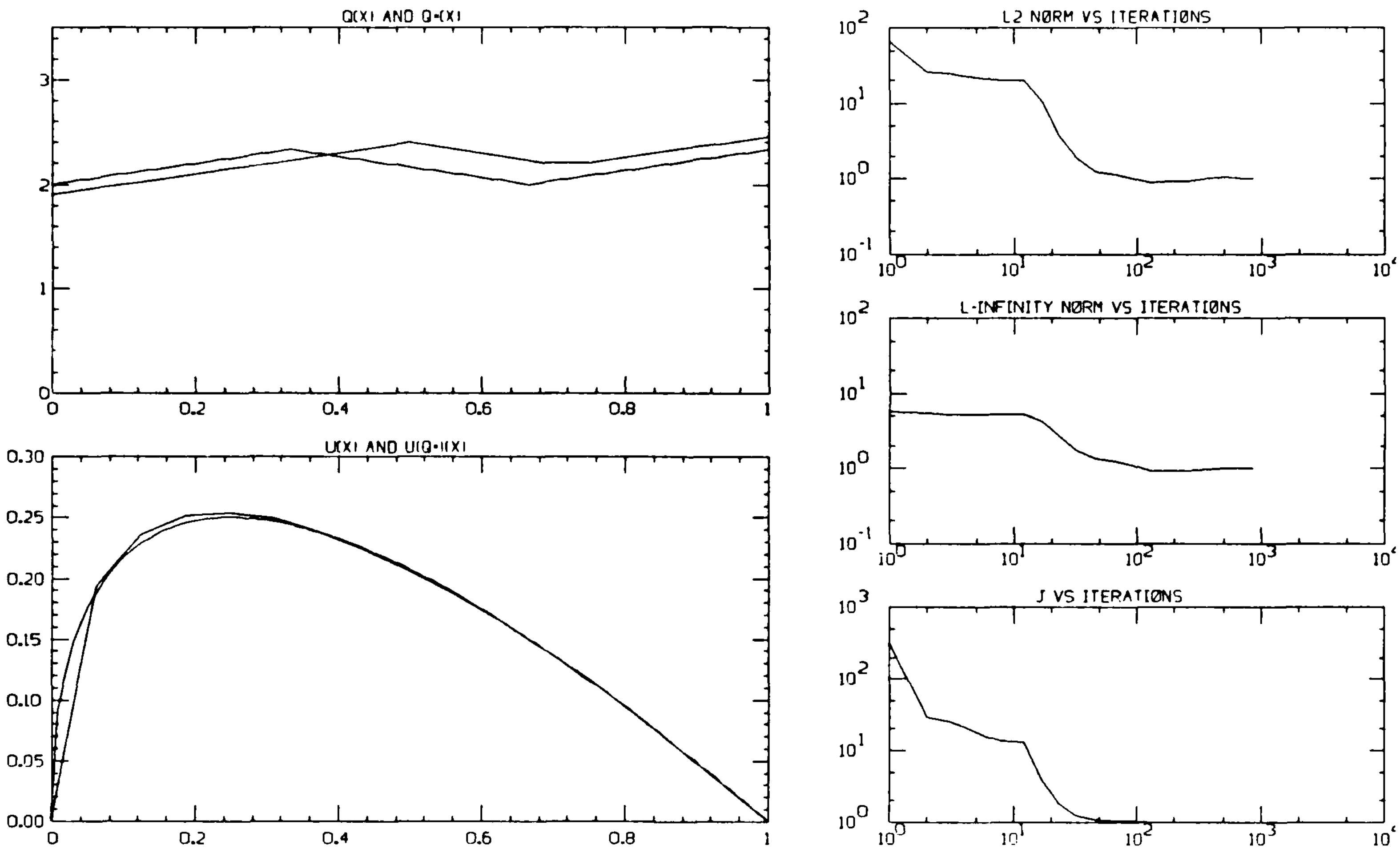
Fig 11: Unconstrained Estimation #2: N=96 M=32



ITERATIØN 9999 J = 0.309173D-05

L2 NØRM Q*-Q = 0.309931D-02 L-INFINITY NØRM Q*-Q = 0.353348D+00

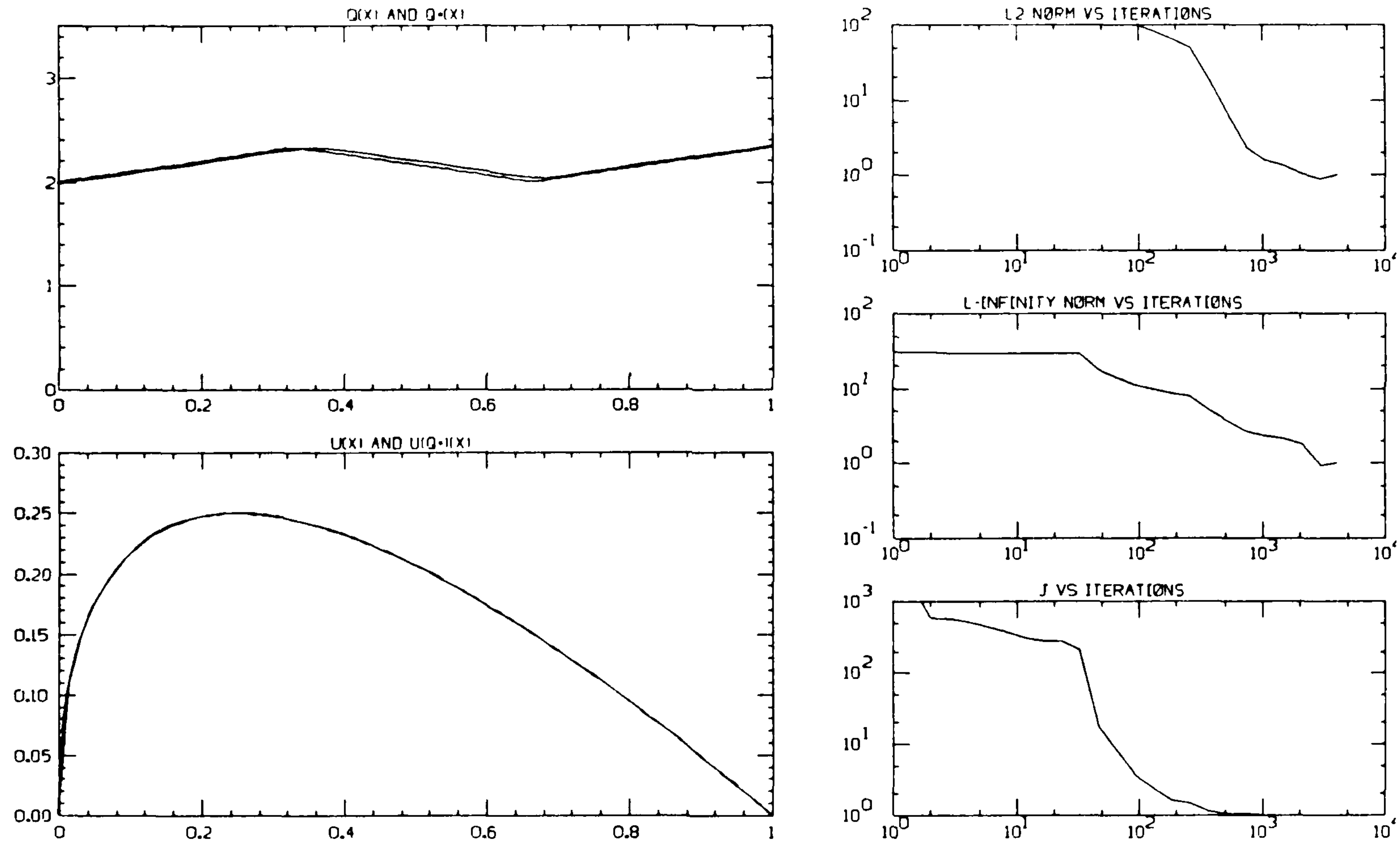
Fig 12: Constrained Estimation #2: N=16 M=16



ITERATIØN 851 J = 0.120547D-03

L2 NØRM Q*-Q = 0.209685D-01 L-INFINITY NØRM Q*-Q = 0.232770D+00

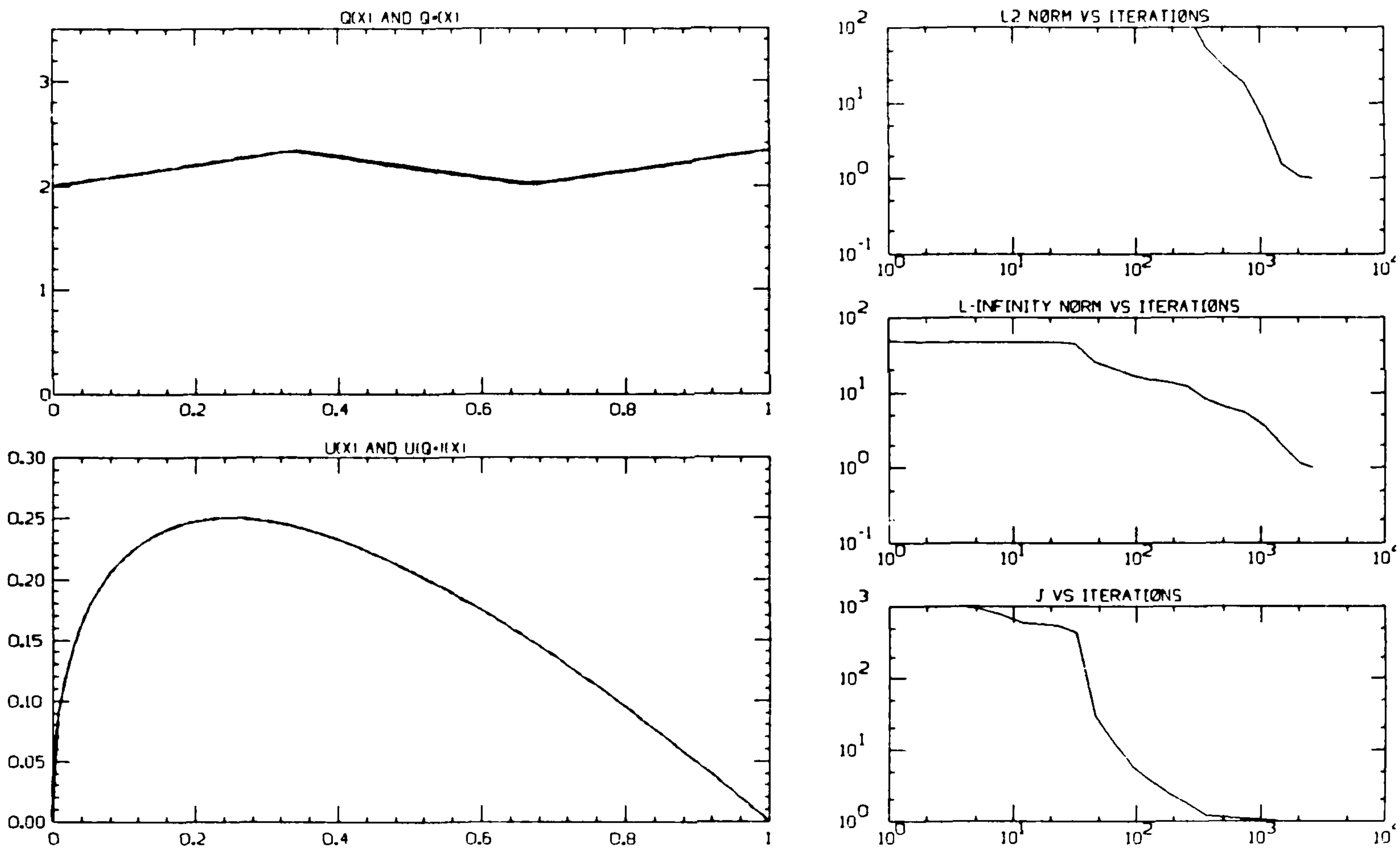
Fig 13: Constrained Estimation #2: N=64 M=32



ITERATIØN 3922 J = 0.695093D-05

L2 NØRM Q*-Q = 0.606869D-03 L-INFINITY NØRM Q*-Q = 0.426997D-01

Fig 14: Constrained Estimation #2: N=96 M=32



ITERATIØN 2534　　J = 0.352755D-05

L2 NØRM Q*-Q = 0.177783D-03　　L-INFINITY NØRM Q*-Q = 0.273098D-01

Finding an Appropriate Value for L

In the previous section we choose the constraint parameter L to be one, which was the slope of the true parameter. In general the maximum slope of the true parameter will not be known exactly, but must be estimated. Figure 15 shows the behavior of the estimate, q^M, for Example 2 as L is increased. As the constraints are eased, the estimate approaches the true parameter (graphed with a dashed line here and in subsequent figures), then as the constraints are eased further the estimates develop the oscillations characteristic of the unconstrained algorithm. From the graph the best value of L can be seen, provided you have some apriori knowledge of the oscillations which are present in the true parameter.

These examples are slightly artificial in that the slope of q(x) is always one, thus there is a value of L which is exactly the true magnitude of the slope everywhere. This is not the case for Example 5 which starts and finishes with slope one, but is parabolic for the middle third. Figure 16 shows the change in estimates for this example as L is increased, while Figure 17 shows the change in estimates as M is increased. As can be seen the behavior is similar even though the constraints are not exact everywhere.

Fig 15 : Constrained Estimates #2 N=64 M=16

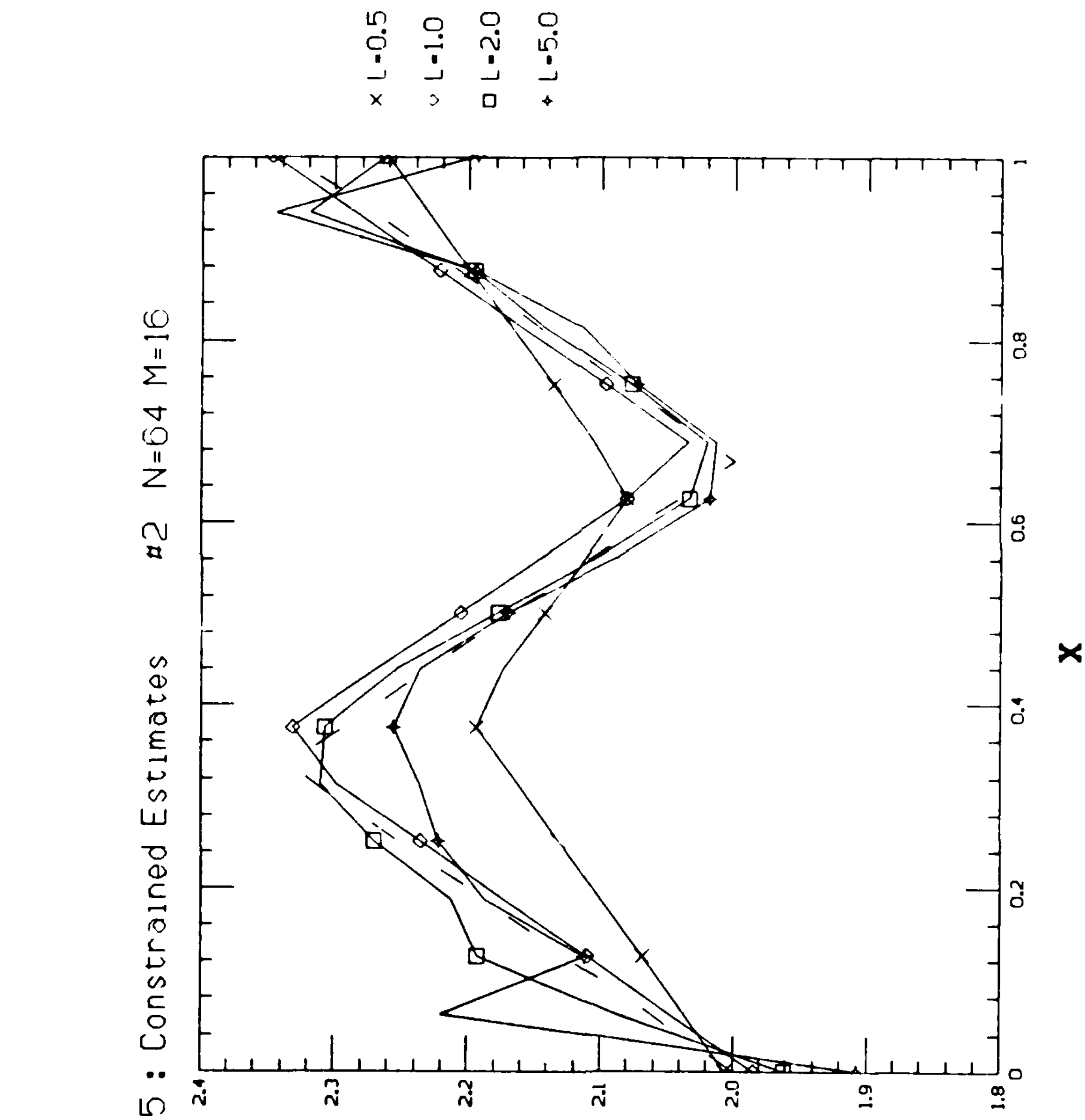


Fig 16 : Constrained Estimates #5 N=64 M=16

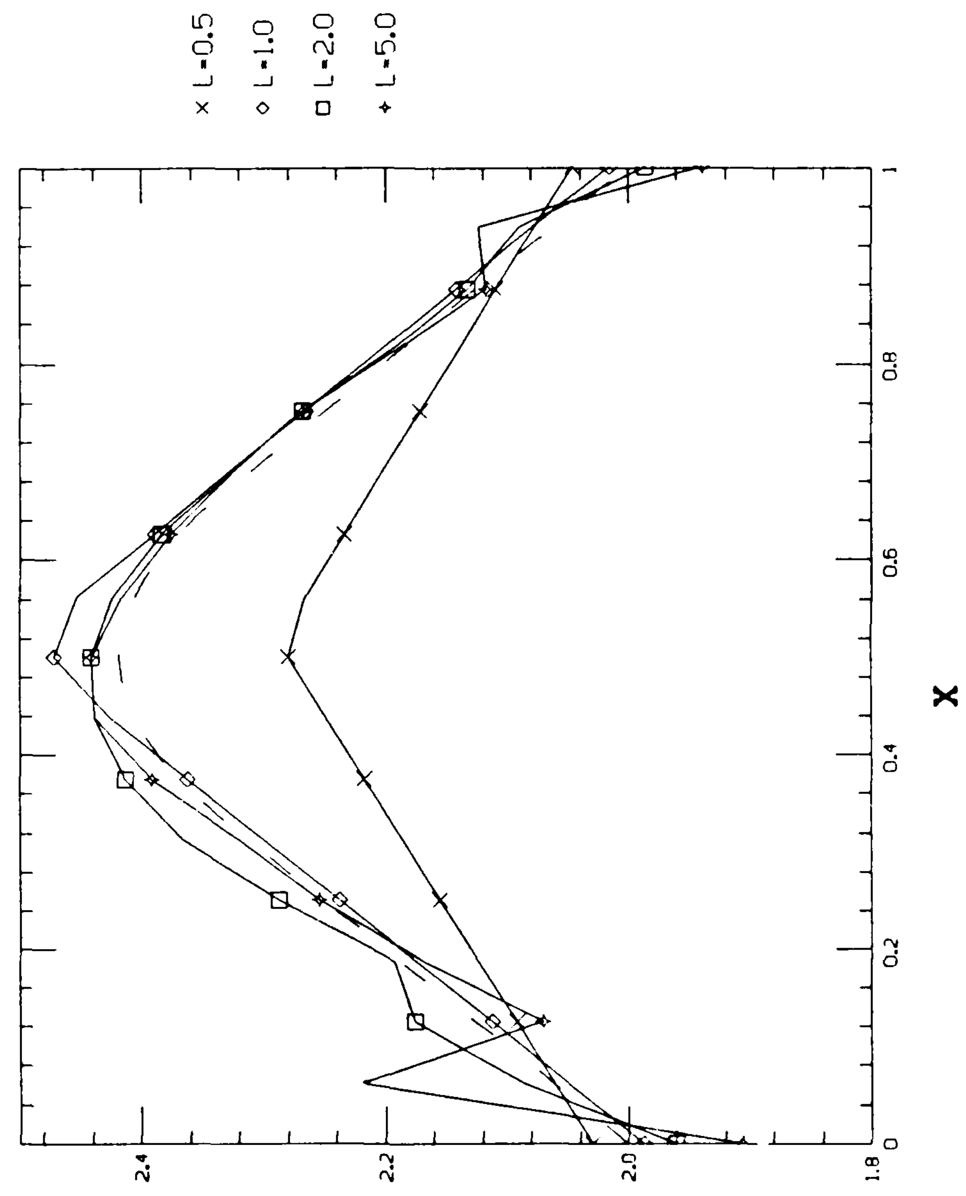
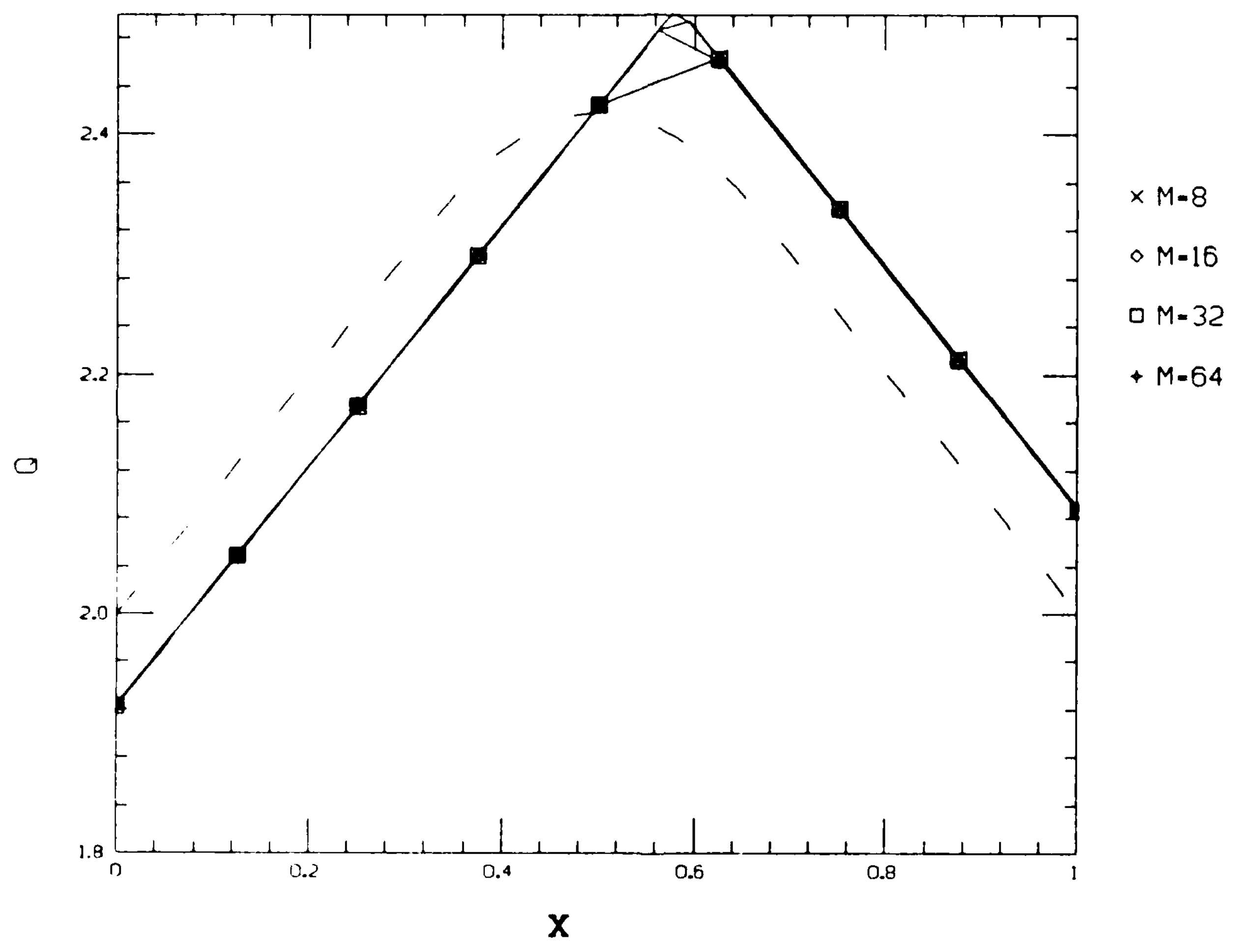


Fig 17 : Constrained Estimates (L=1) #5   N=16



Finding Appropriate Values for α and β

Some discusion of the problem of finding a suitable regularization parameter β is given in [5]. We found a suitable β by trying a wide range of values. Figure 18 shows the estimates for various values of β. It can be seen that as β is decreased the estimates converge towards the true answer and then start to develop the oscillations typical of the unconstrained method. From the graph the best value of β can be seen, without knowing the true parameter, provided some assumptions are made about the sort of oscillations that are present in the true parameter.

Figure 19 shows the L^{∞} norm of the error versus β for several values of α. Note that the estimate is much less sensitive to the value of β as α is decreased. Reducing α has the effect of limiting the slope of the estimate more than the absolute value of the function.

For the work in this paper, comparing the various algorithms, we used our knowledge of the true q to choose a suitable value of β.

Fig 18: Tikhonov Estimates #2: N=64 M=16

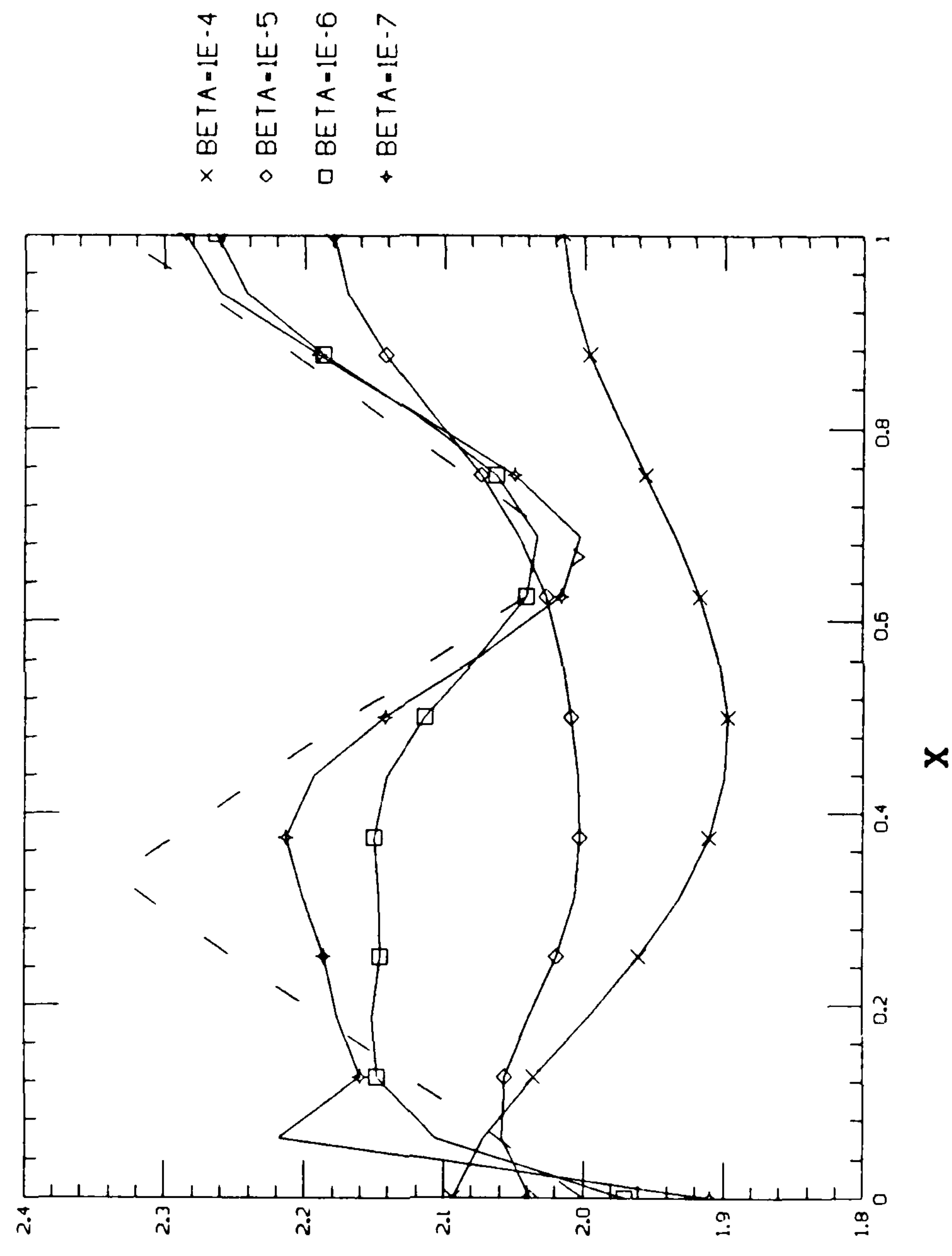
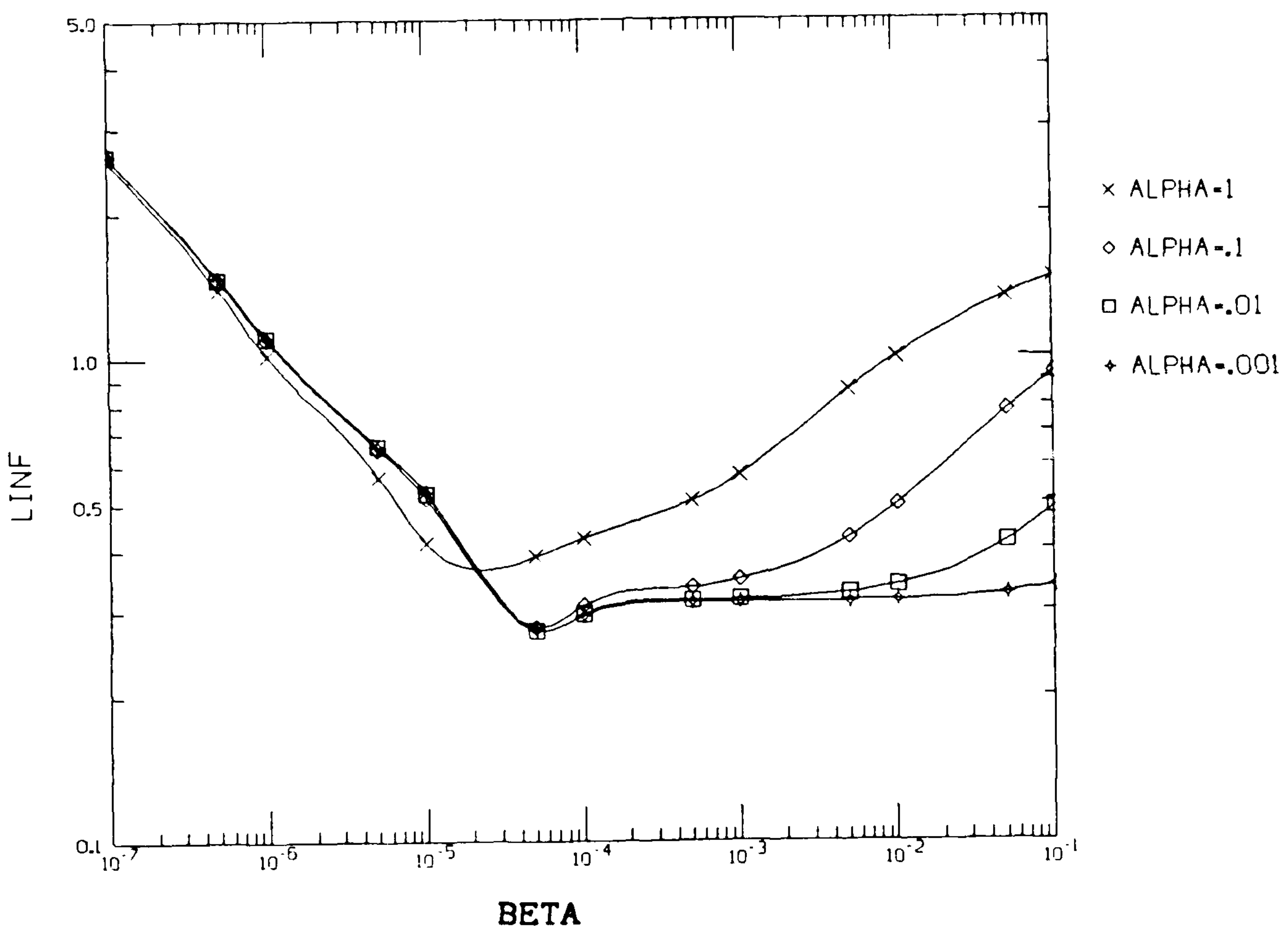


Fig 19: The Effect of Changing Alpha #2: N=8 M=15



The Tikhonov versus Constrained Estimation Algorithms

For low N the behavior of the Tikhonov estimation algorithm is similar to that of the constrained algorithm. For Example 2 both are stable as M is increased (Fig 20) and give similar estimates, provided β, α and L are suitable values. This is in contrast to the unconstrained estimates, which tend to develop oscillations as M is increased. However with low N both algorithms give an estimate with no real detail (Figs 21 and 22).

As N is increased while holding M fixed the estimates for all three algorithms improve (Fig 23), however the constrained algorithm performs best, while the Tikhonov algorithm tends to produce a flattened estimate (Fig 24).

The bias in the Tikhonov method, evidenced by an estimate which is somewhat flatter than the true parameter, is more marked the more strongly varying q^M is. Figure 25 shows the improvement in the estimate as N is increased for Example 3. The constrained algorithm performs much better than the Tikhonov algorithm which produces a badly biased estimate, even for the best β, chosen knowing the true parameter. When β is reduced, to reduce the bias, oscillations develop in the estimate before it gets near to the true parameter (Fig 26).

The constrained estimation algorithm does not have this problem. As L is increased the estimates come close to the true parameter and only then develop oscillations (Fig 27).

For a very flat function the Tikhonov method works well. The parameter q in Example 4 is just a constant function. As β is decreased, with sufficiently large N, the Tikhonov estimates converge to the true parameter and only then start to develop oscillations as β is decreased further (Fig 28).

For this flat function the constrained algorithm also works well, if given much tighter constraints than in the previous examples. The estimates for various values of L are shown in Figure 29.

Fig 20 : Linf vs M #2 : N=8

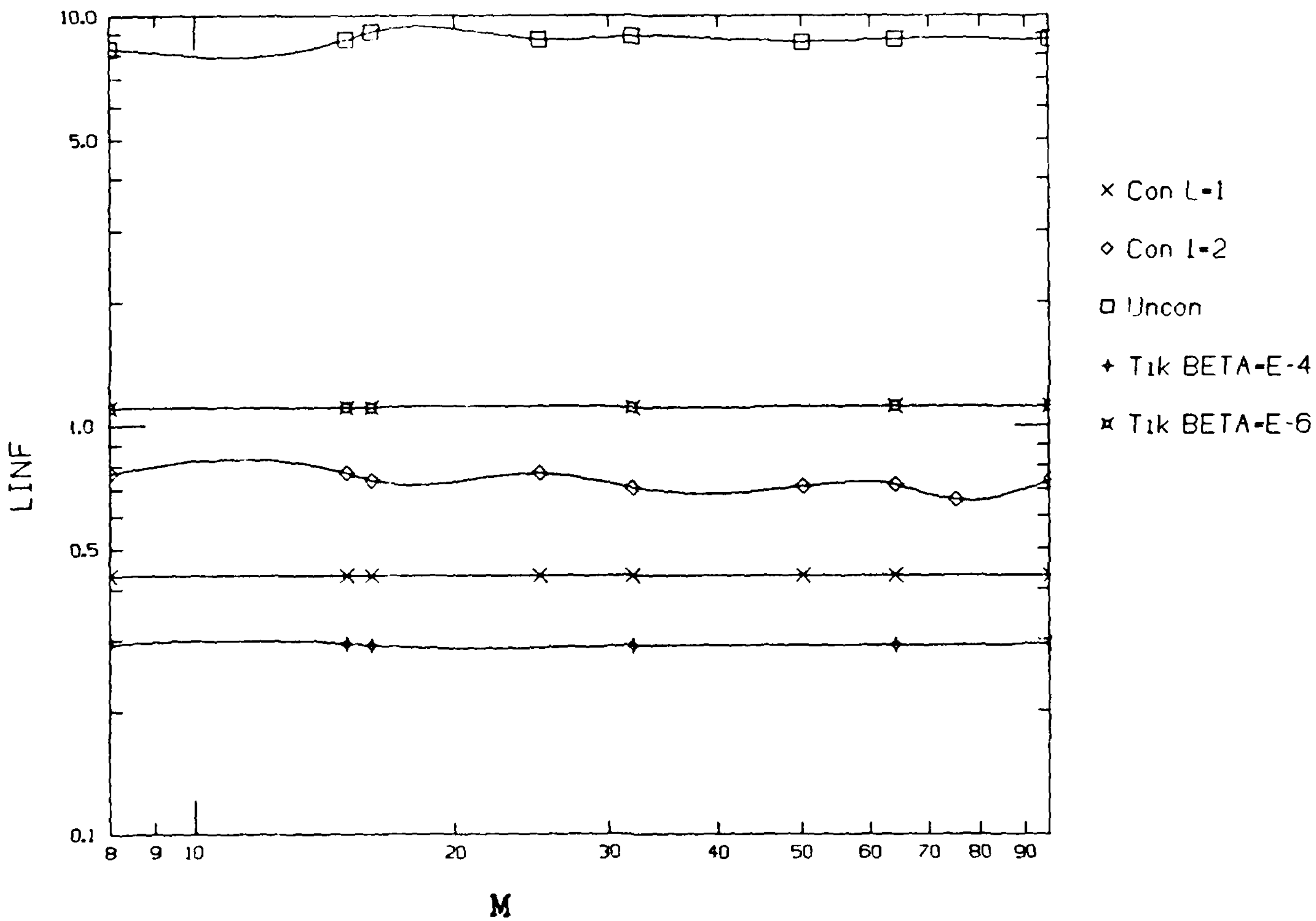


Fig 21 : Constrained Estimates of Q   #2: N=8 M=15

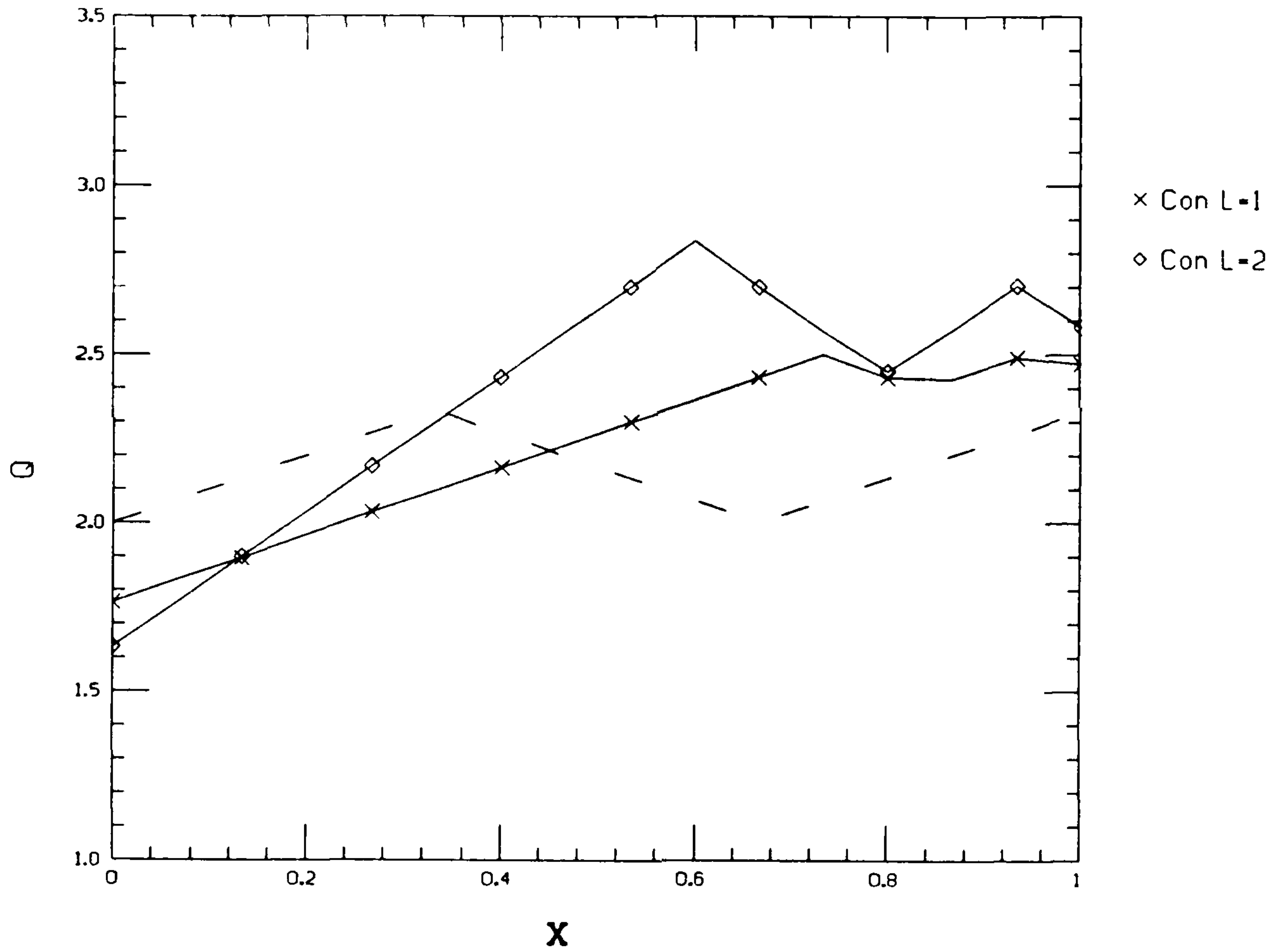


Fig 22 : Tikhonov Estimates of Q #2: N=8 M=15

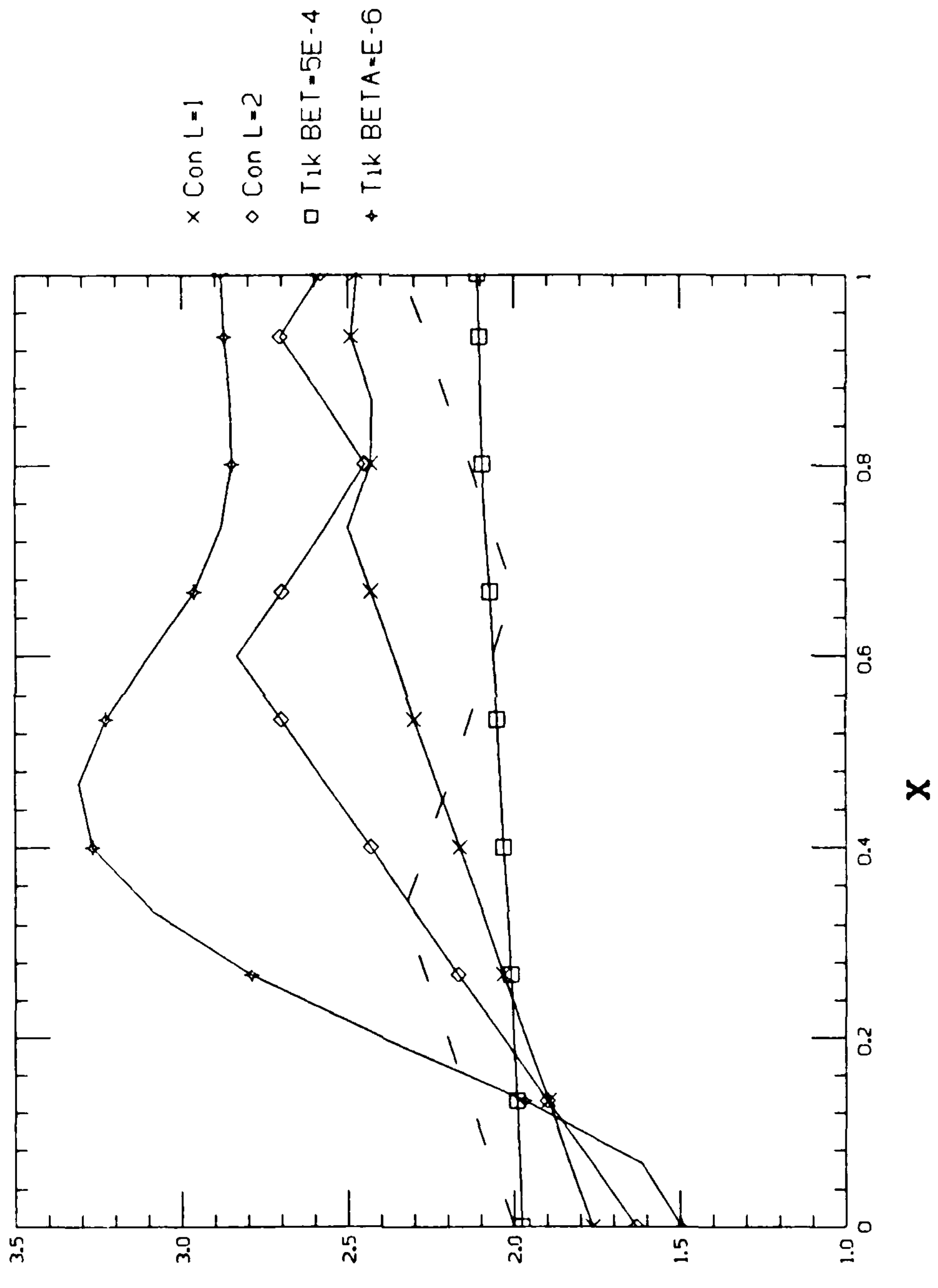
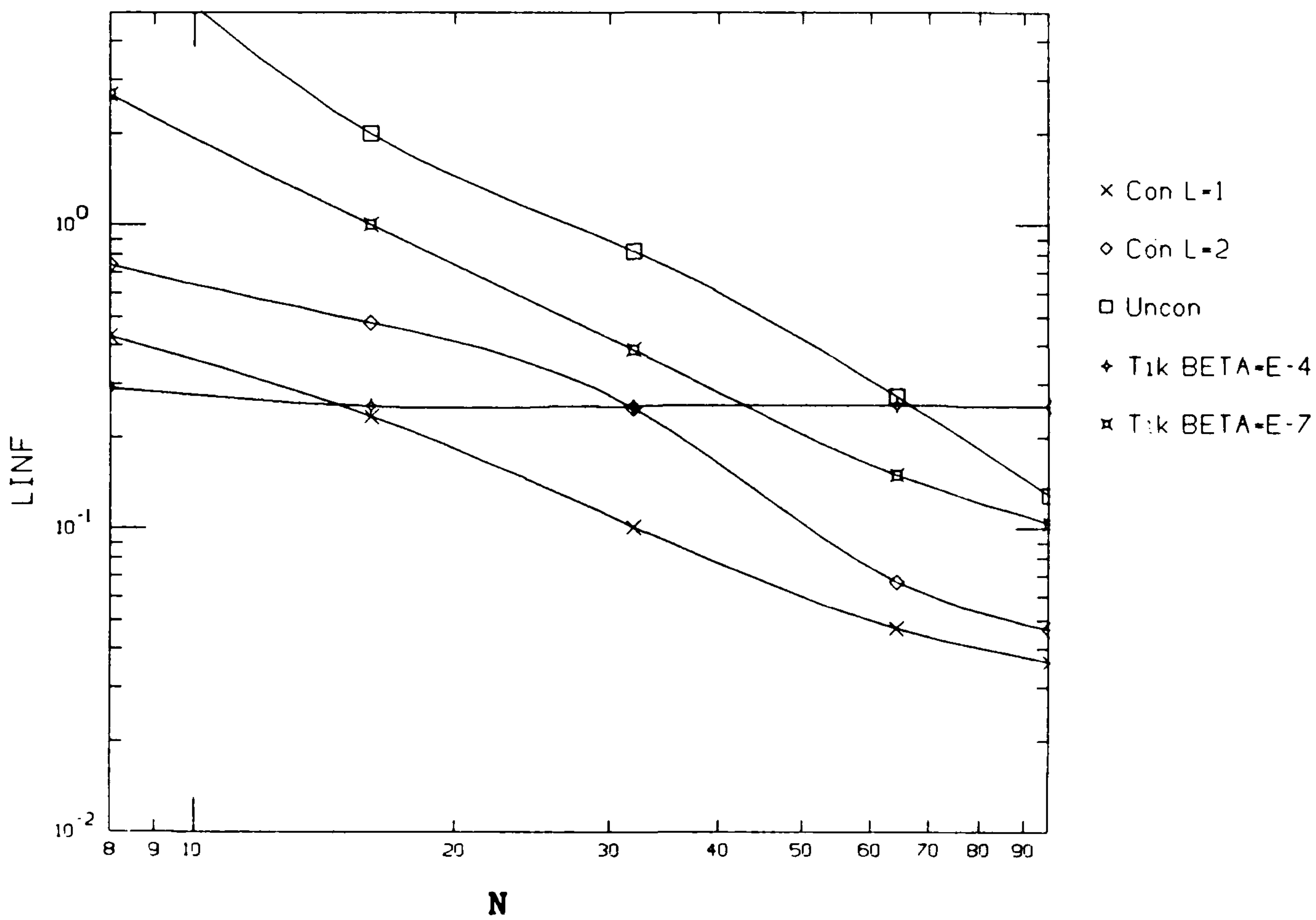


Fig 23 : Linf vs N #2 M=16



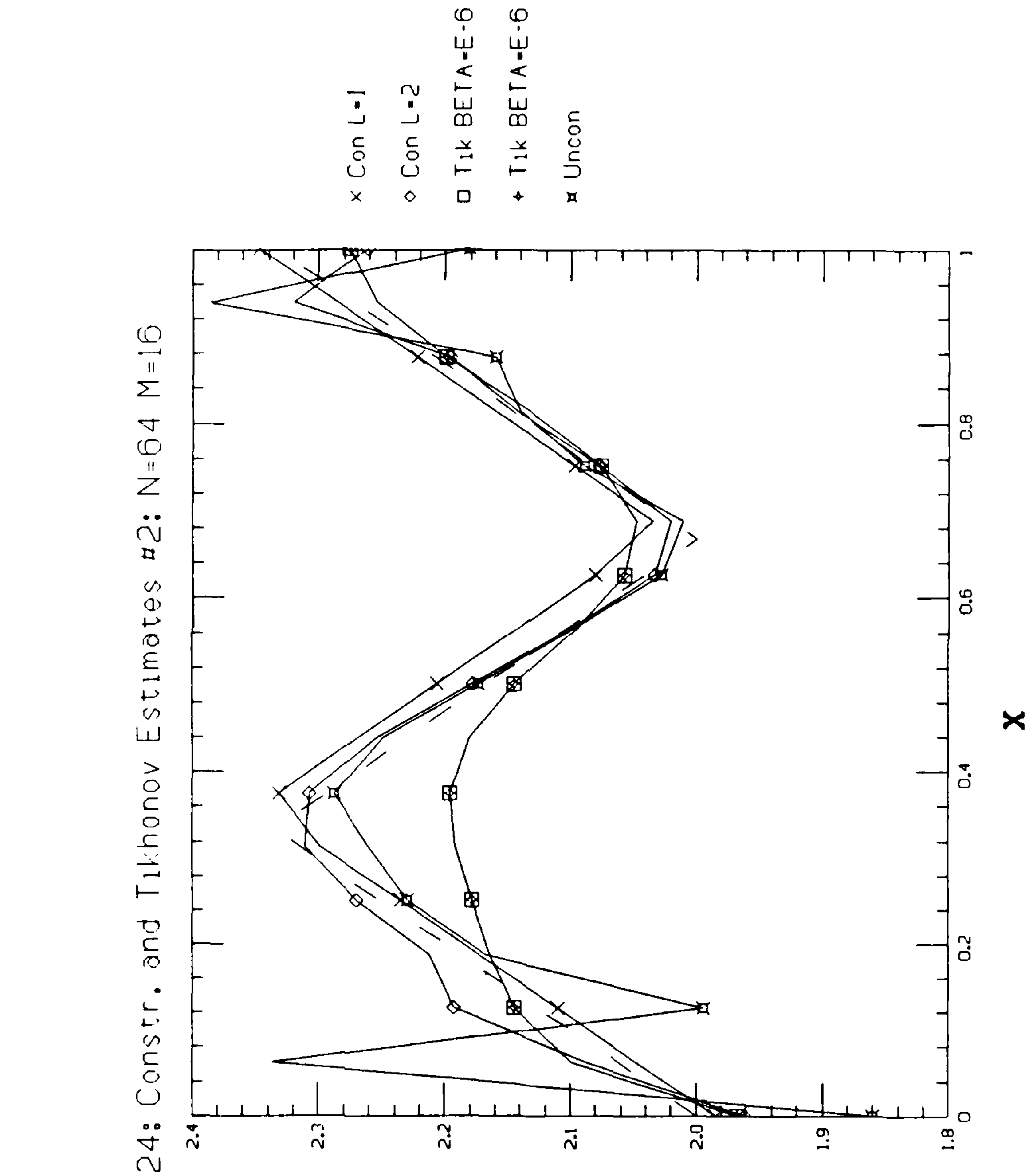
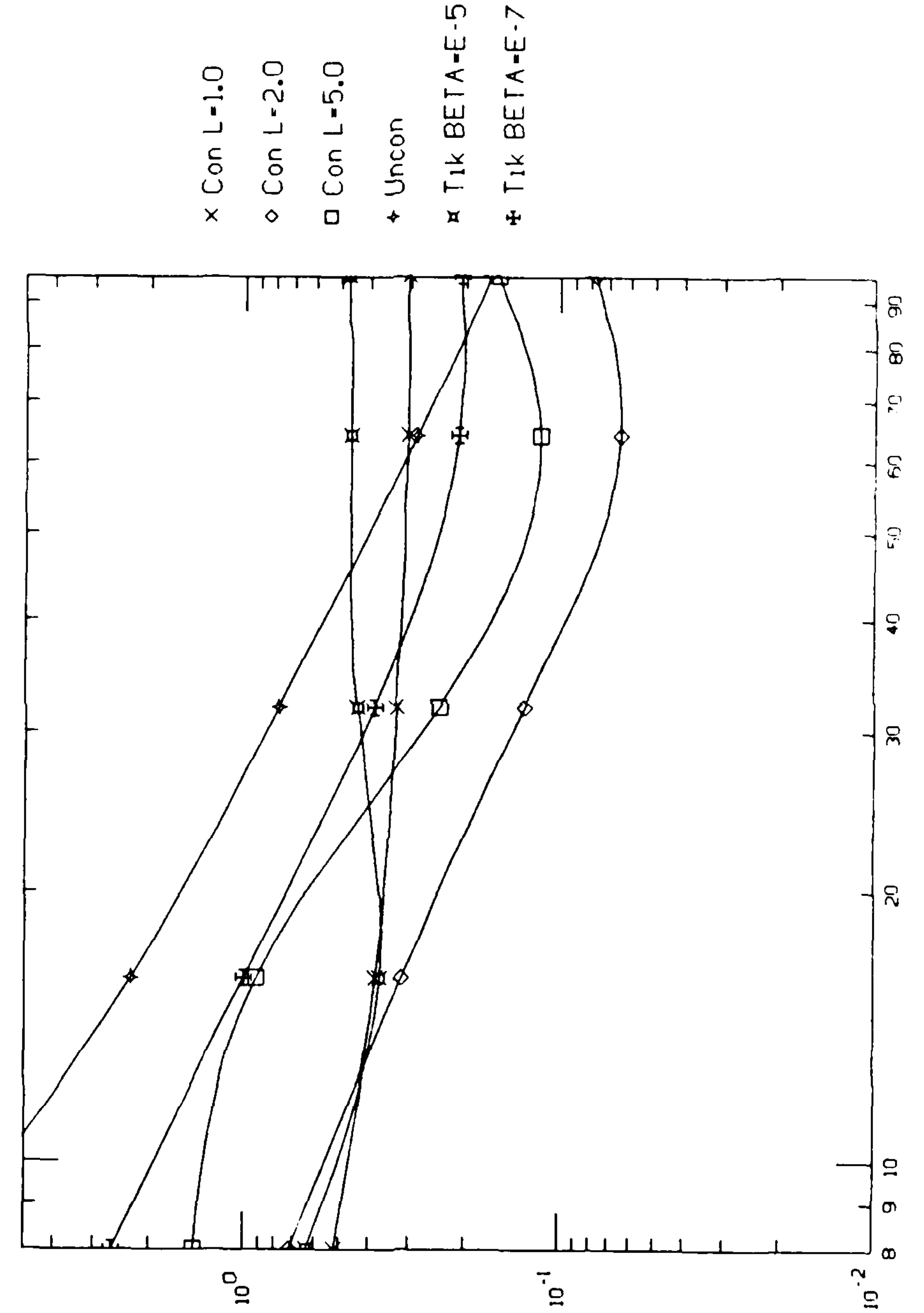


Fig 24: Constr. and Tikhonov Estimates #2: N=64 M=16

Fig 25: Linf vs N



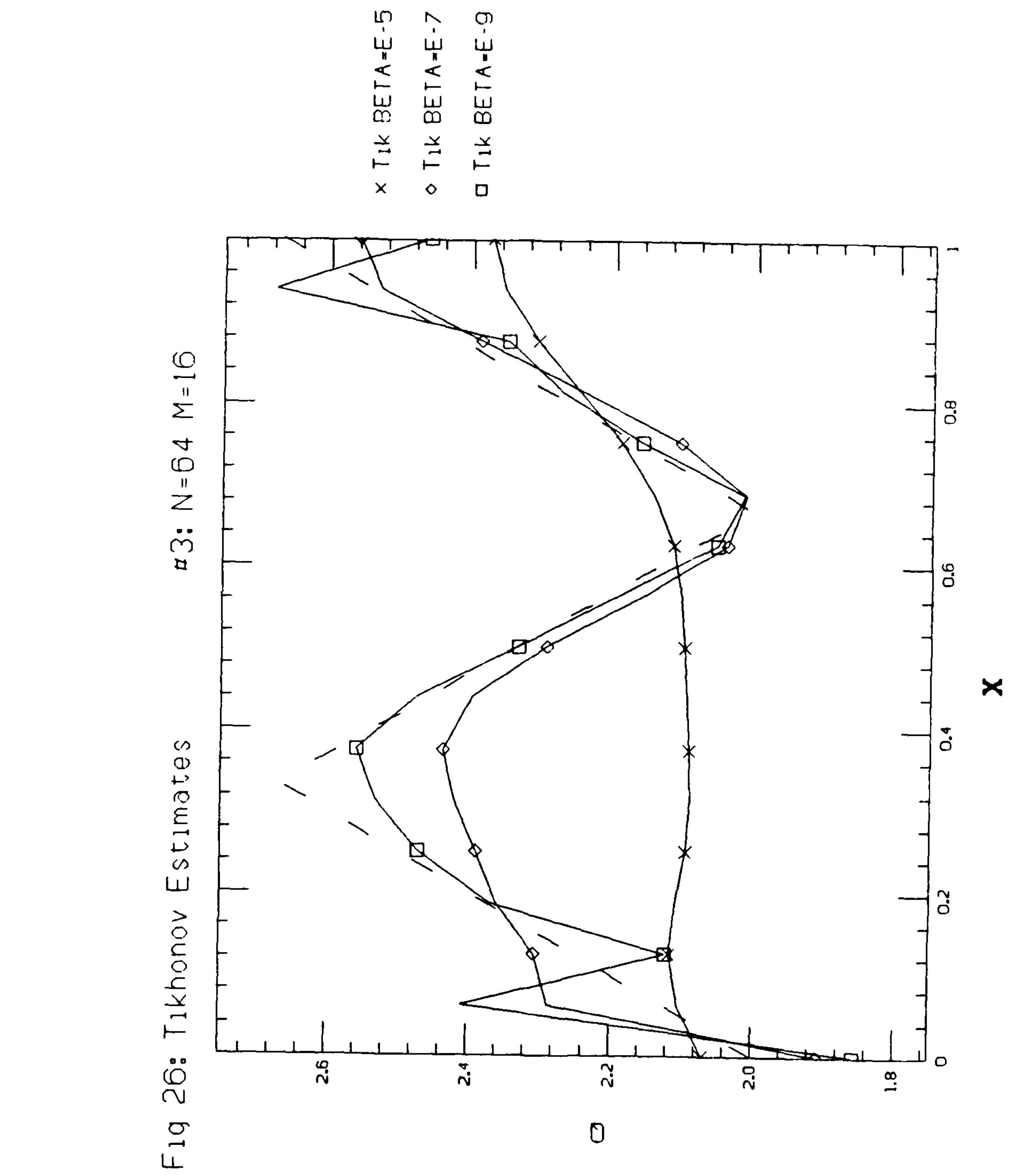


Fig 26: Tikhonov Estimates #3: N=64 M=16

Fig 27: Constrained Estimates of Q   #3: N=64 M=16

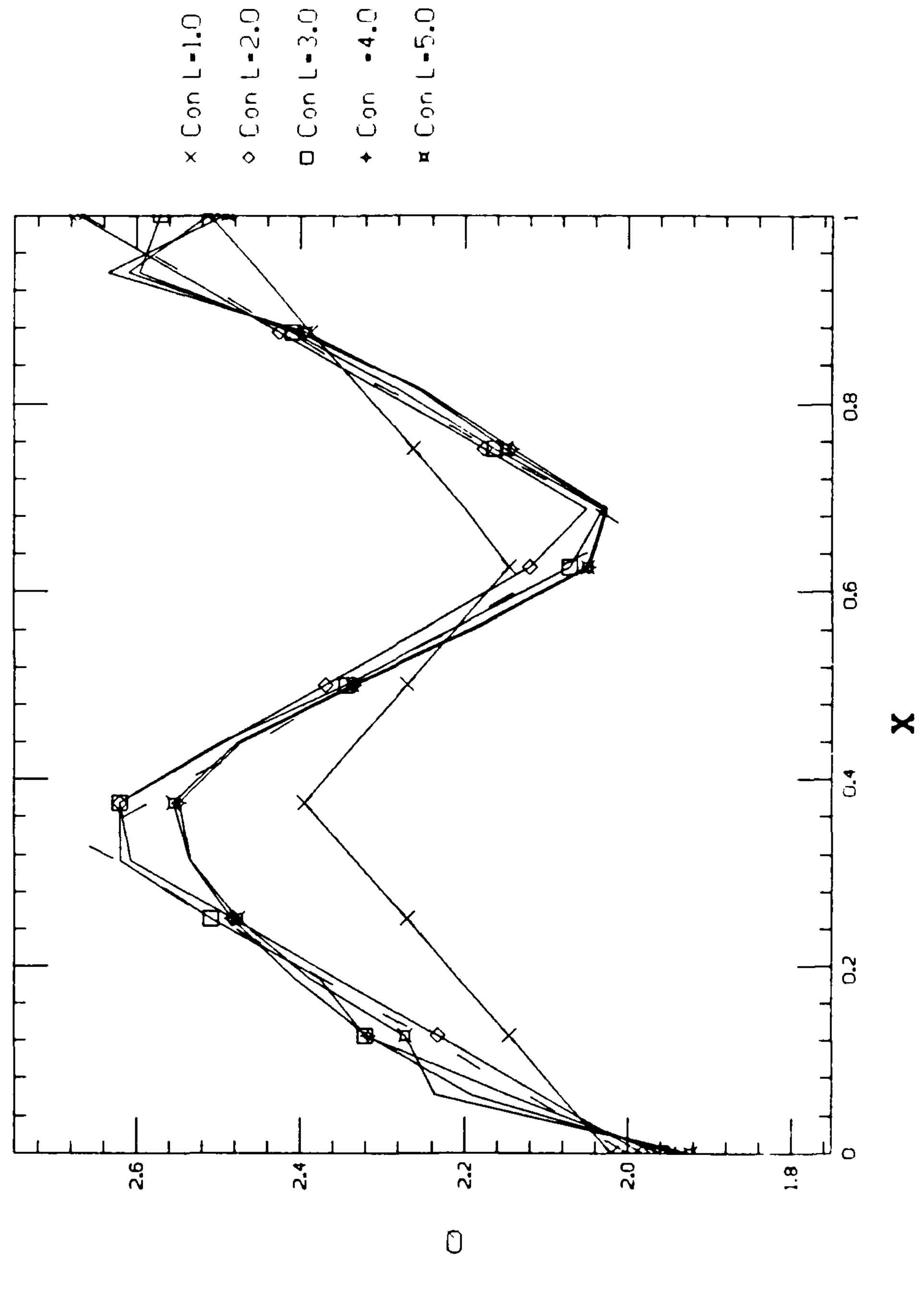


Fig 28: Tikhonov Estimates of Q #4: N=64 M=16

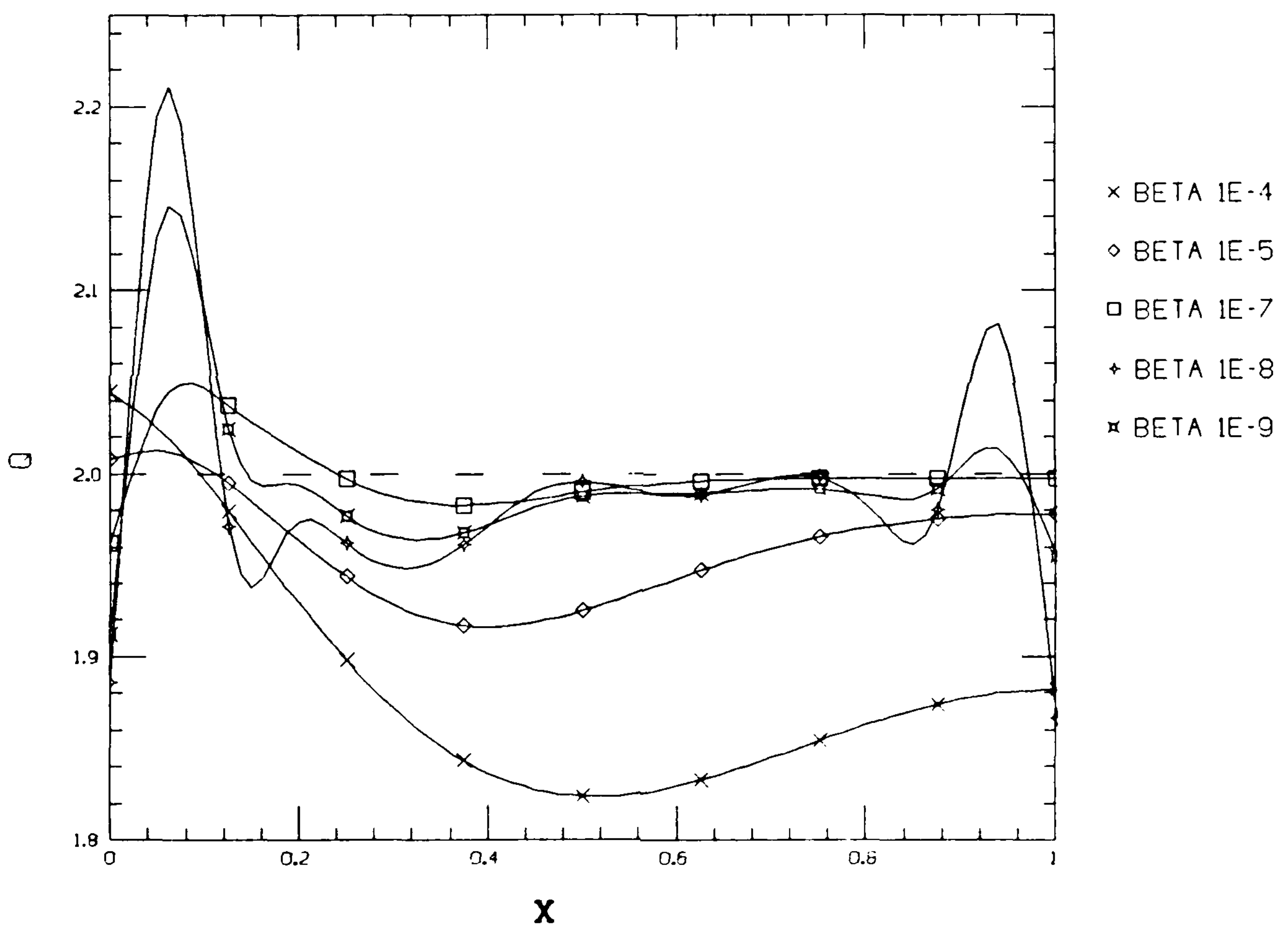
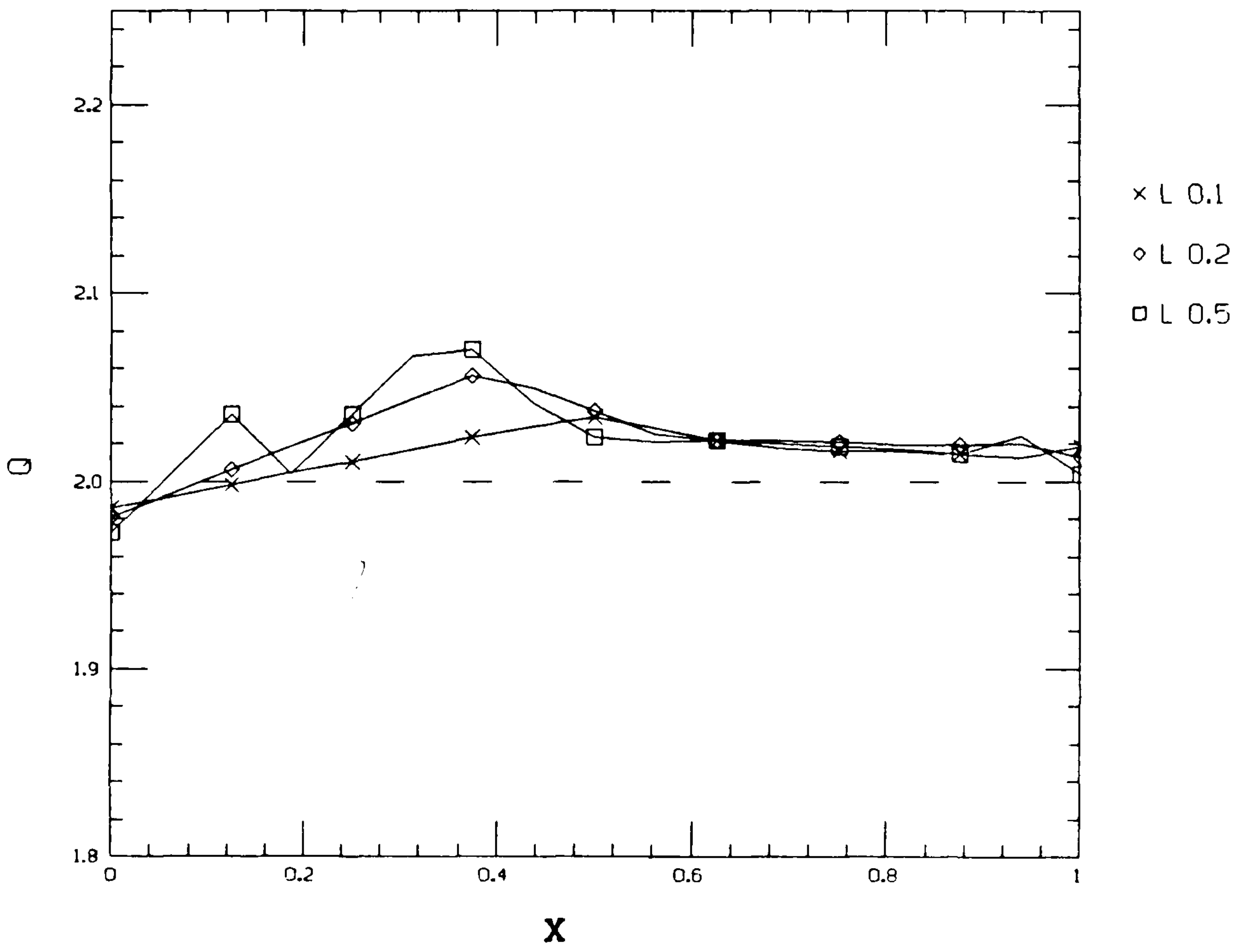


Fig 29: Constrained Estimates of Q #4: N=64 M=16



The Stability of the Estimates With Perturbations of U

To test the stability of our algorithms with respect to noise in the data, we took the true u,q and f in Example 2, but perturbed u to produce the data $\hat{u} = u_p$ in the cost criterion J. We used a perturbation of the form

$$u_p(x) \quad = u(x) + p(x)/K$$

where p(x) is the perturbing function, and K determines the size of the perturbation. As K is increased $u_p(x) \to u(x)$. We used two perturbation functions:

(a) $p_1(x) = x(1-x)$

(b) $p_2(x) = 1$

Note that $p_1(x)$ satisfies the same boundary conditions as u while $p_2(x)$ does not.

The unconstrained, constrained and Tikhonov estimates for several values of K with perturbation function $p_1(x)$ and $N=8$ and $M=15$ are shown in Figures 30 – 32. With the constrained and Tikhonov algorithms the estimates improve steadily as the error in the input data, u, is reduced, while the unconstrained estimates are bad even with the exact data. The L^∞ norm of the errors in the final estimates versus K for $N=8$ $M=15$, $N=16$ $M=16$ and $N=64$ $M=16$ for both perturbations are shown in Figures 33 through 38. Figure 33 also shows the results for the unconstrained algorithm when weaker convergence criteria are used.

For all the values of N the behavior of the constrained and Tikhonov methods are similar, with the estimate improving steadily as $u_p(x) \to u(x)$, the true solution.

For small N, the unconstrained estimates actually get worse as u_p approaches u(x), and for all perturbations these estimates are worse than the initial guess, which has an error in the L^∞ norm of 4/3. For large N the unconstrained estimates are stable with respect to perturbations of the input data.

By comparing Figures 35 and 38 it can be seen that perturbation 2, which does not satisfy the same boundary conditions as u, has a much greater effect on the estimate error.

Fig 30: Constr. Est. of Q, Pert #1 #2: N=8 M=15

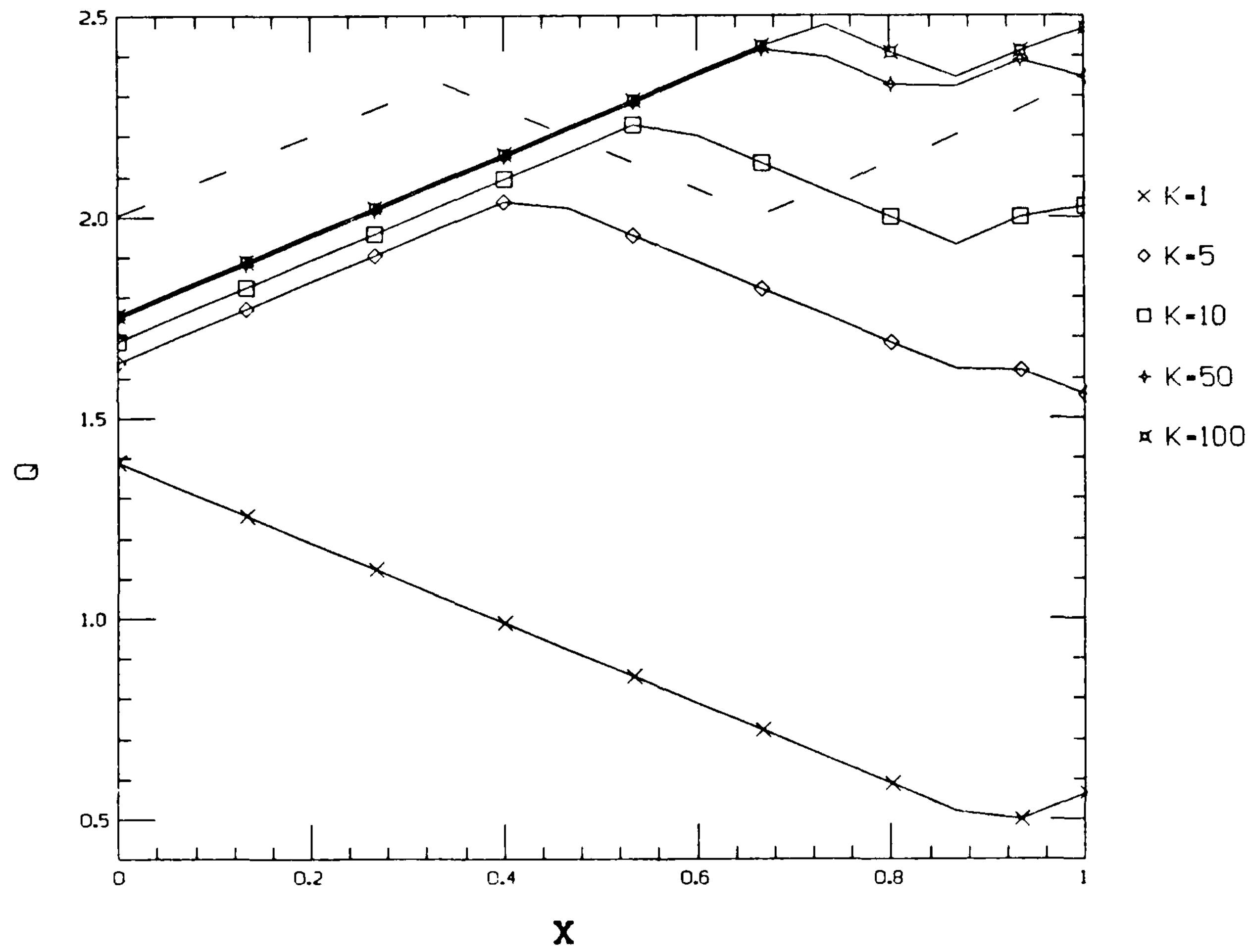
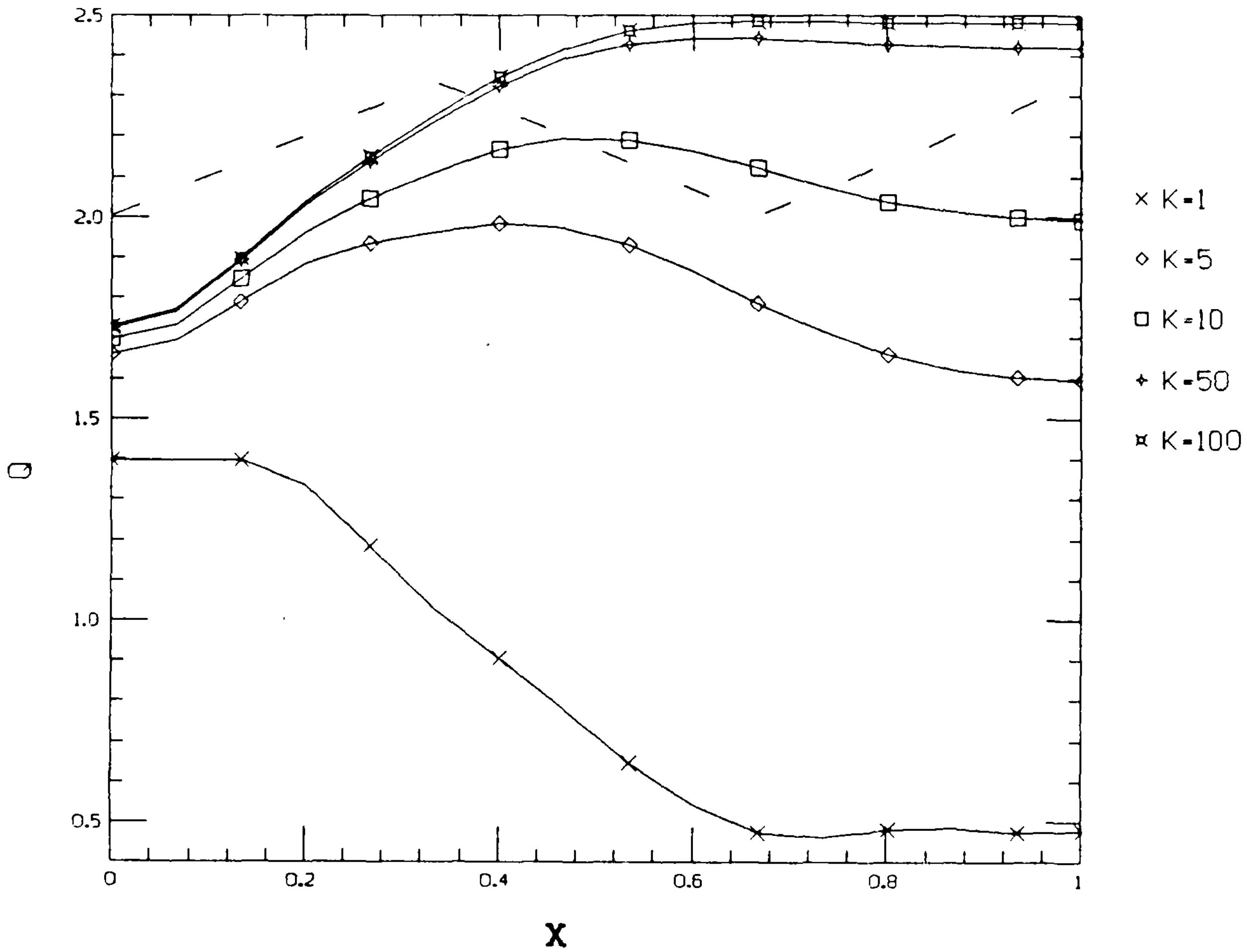


Fig 31: Tik Est Of Q, Pert #1 #2: N=8 M=15



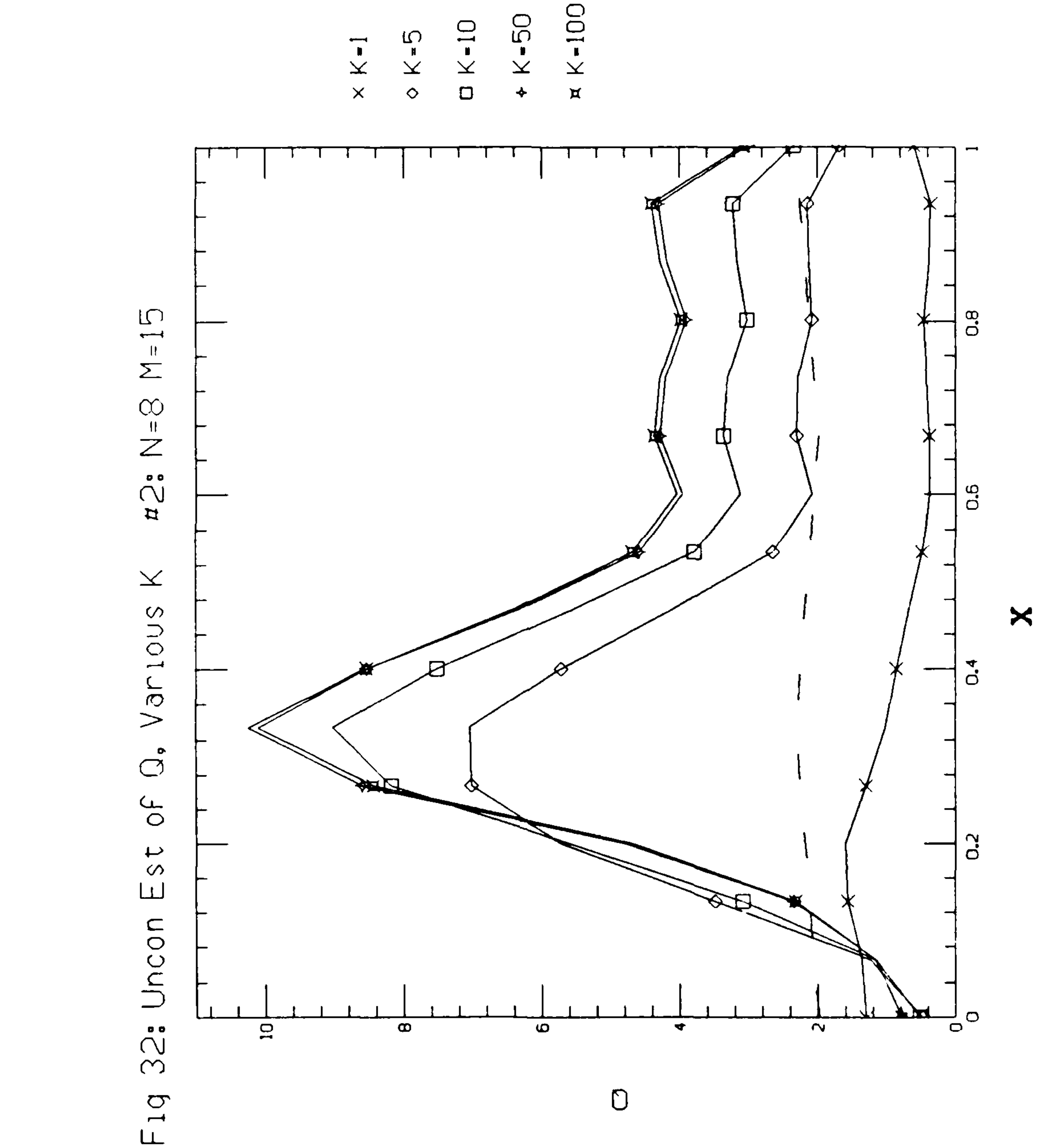


Fig 32: Uncon Est of Q, Various K #2: N=8 M=15

Fig 33: Linf vs K   Perturbation #1   #2: N=8 M=15

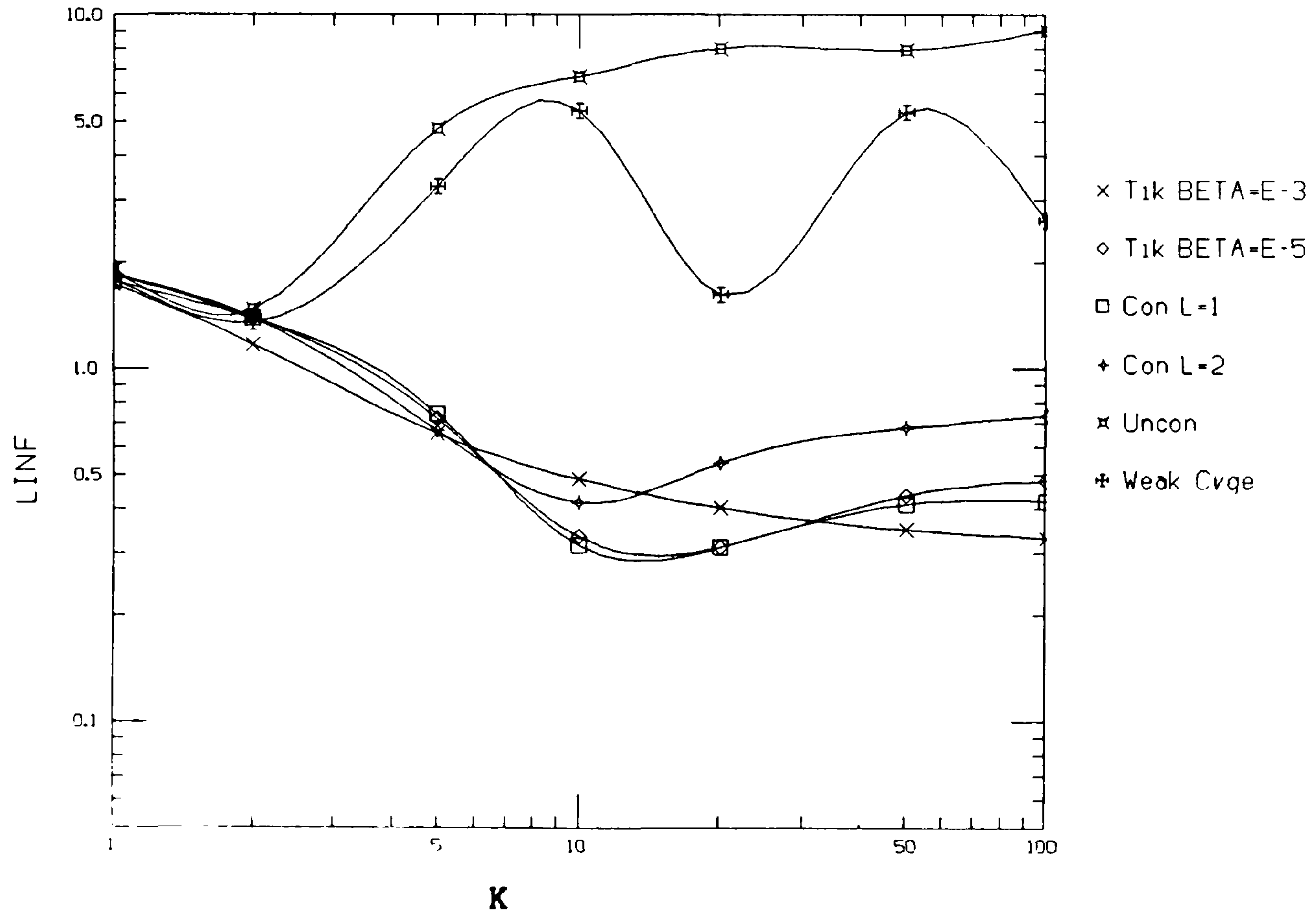


Fig 3.4: Linf v. K   Perturbation #1   #2: N=16 M=16

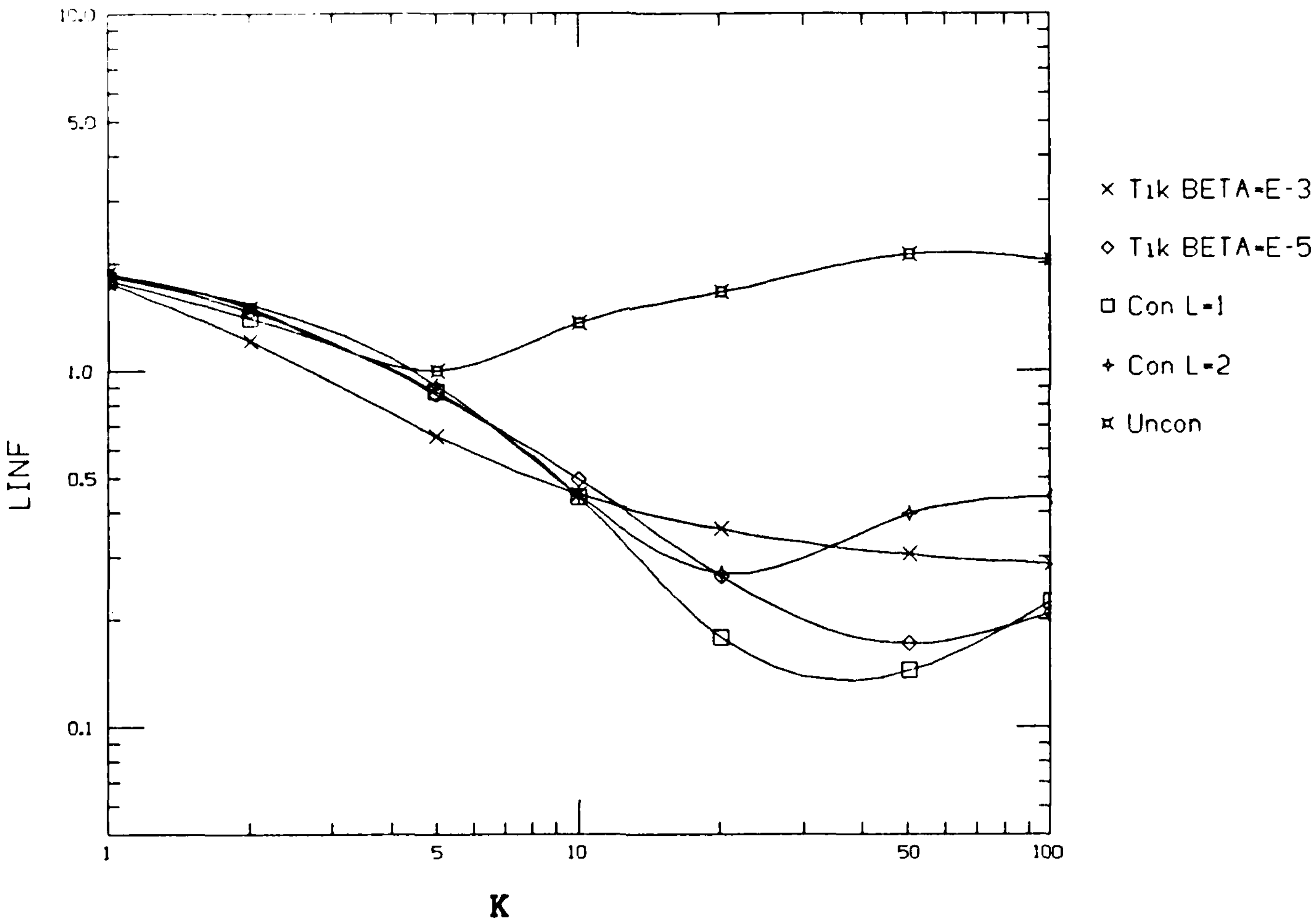


Fig 35: Linf vs K   Perturbation #1   #2: N=64 M=16

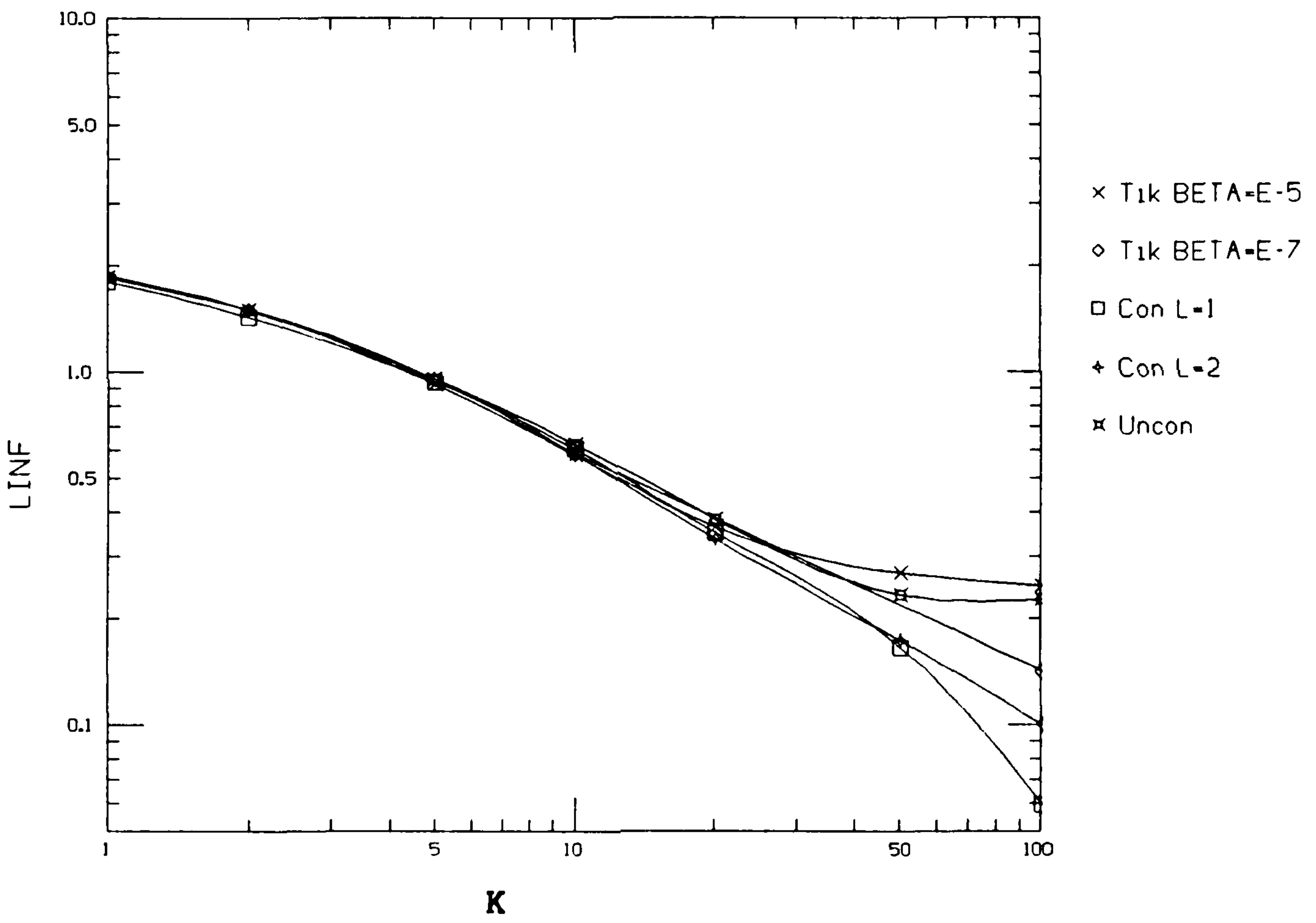


Fig 36: Linf vs K, Perturbation #2   #2: N=8 M=15

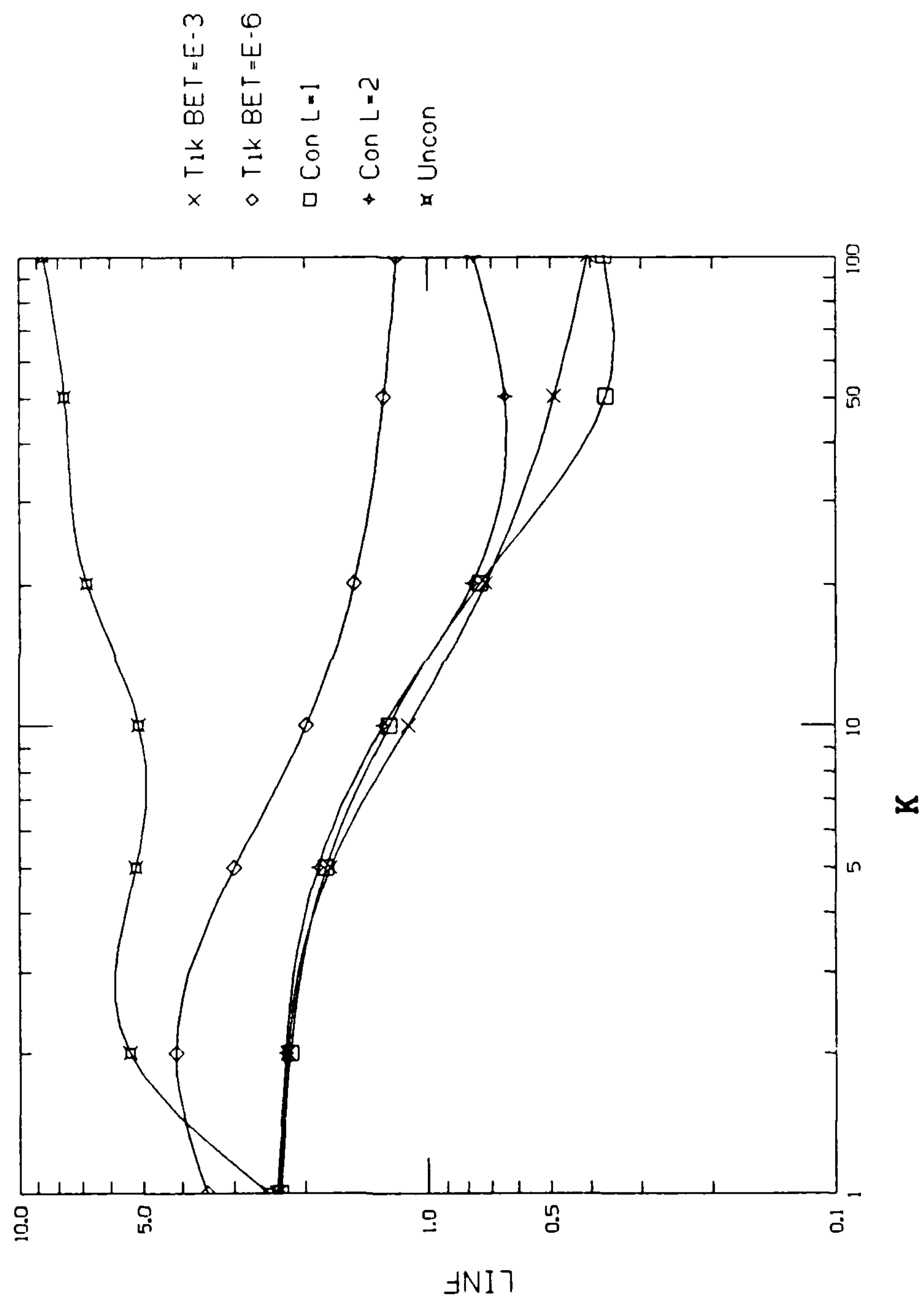


Fig 37: Linf vs K   Perturbation #2   #2: N=16 M=16

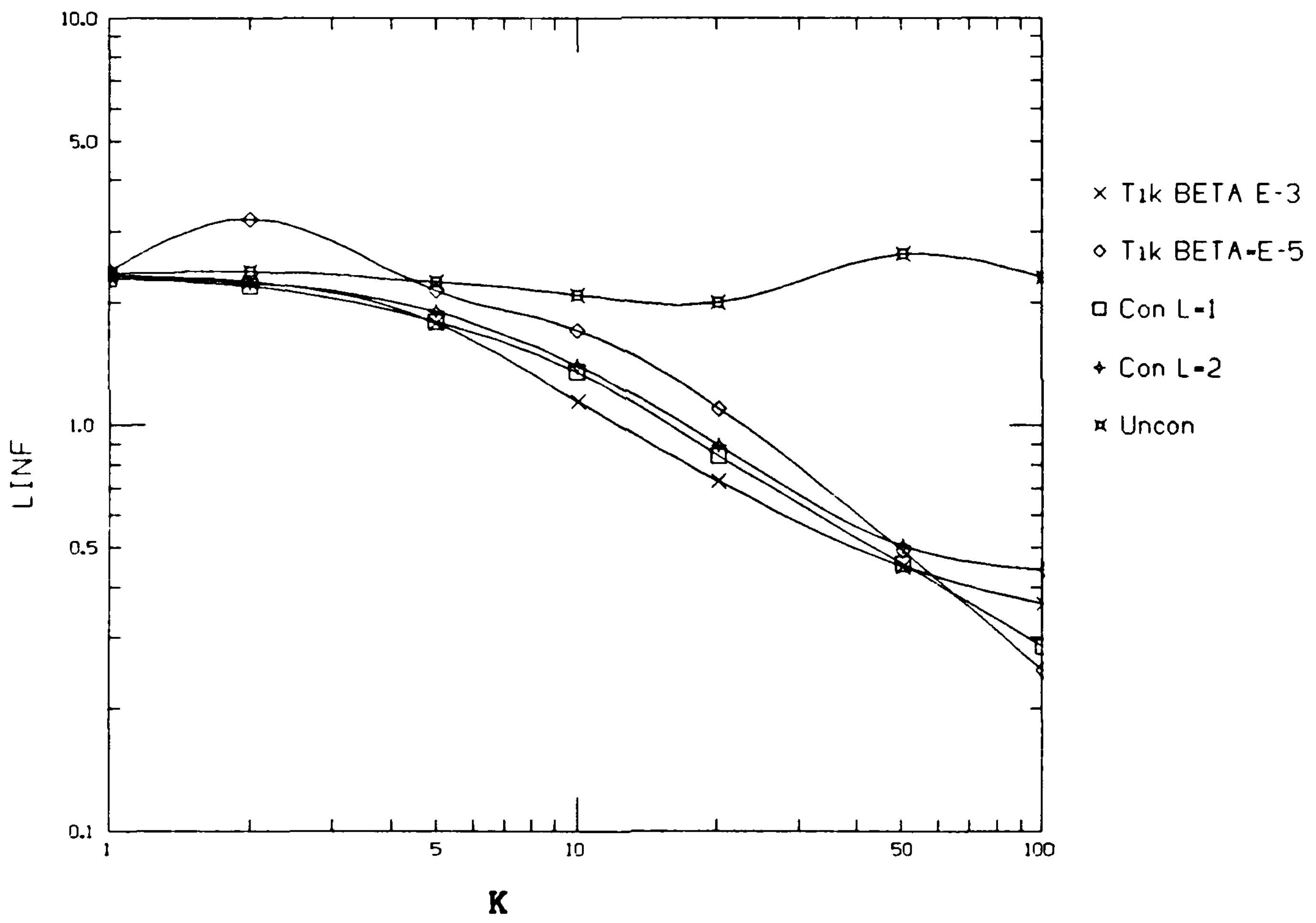
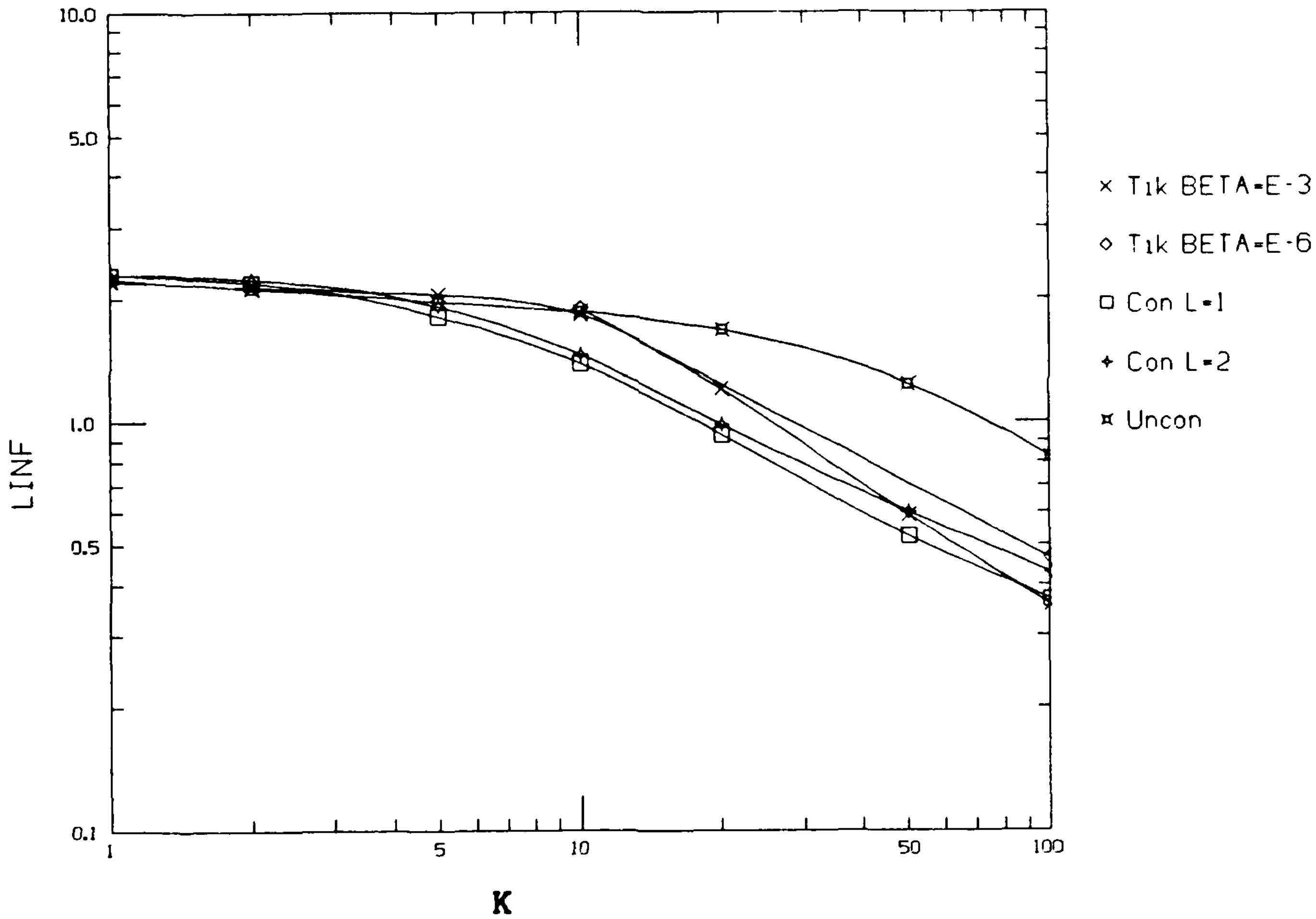


Fig 38: Linf vs K Perturbation #2 #2: N=64 M=16



The Convergence Criteria

In finding the estimates the reduced gradient algorithm was run until one of the following conditions was met:

(a) the sequences q_i^M and J_i both converged, where x_i was said to have converged if :

(i) $\|x_i - x_{i-5}\| / \|x_i\| \leq 10^{-5} = \epsilon$

or (ii) $\|x_i - x_{i-5}\| \leq 10^{-14}$

where $\|x\|$ is the L^2 norm squared if x is q^M and is just $J(q^M)$ if x is $J(q^M)$.

(b) the number of iterations exceeeded 999 (or in some cases 9999)

In several of the examples, especially unconstrained examples, the best estimate does not occur when J is minimized. The convergence criteria used in the work for this paper were very tight, and it is possible that in using such tight criteria the optimization of J was taken too far. Thus the oscillations seen in the above examples could be a manifestation of numerical inaccuracies which became significant when reductions in the cost J at each stage of the optimization algorithm were comparable to the error in the evaluation of J.

To check this effect several of the examples where the estimates do deteriorate significantly as J is reduced were rerun with the following weaker convergence criteria.

The reduced gradient algorithm was run until one of the following conditions was met:

(a) the sequences q_i^M and J_i both converged, where x_i was said to have converged if :

(i) $\|x_i - x_{i-5}\| / \|x_i\| \leq 10^{-2} = \epsilon$

or (ii) $\|x_i - x_{i-5}\| \leq 10^{-8}$

where $\| \; \|$ is as described above.

(b) the number of iterations exceeeded 999

Figures 39 through 41 show the results. In general the estimates are slightly better but still show the same defects, just not as highly developed. Even using the knowledge of the true parameter to find the best estimate (best meaning the estimate in the sequence with the smallest error in the L^∞ norm) does not produce a good estimate, just one with the same faults at an earlier stage. We can therefore conclude that the convergence criteria used here, which are stricter than would be used if computing time were significant, are not producing results with errors due to numerical inaccuracies. Rather the deficiencies in the estimates are true manifestations of algorithm weaknesses.

Fig 39: Estimates of Q #2: N=8 M=15

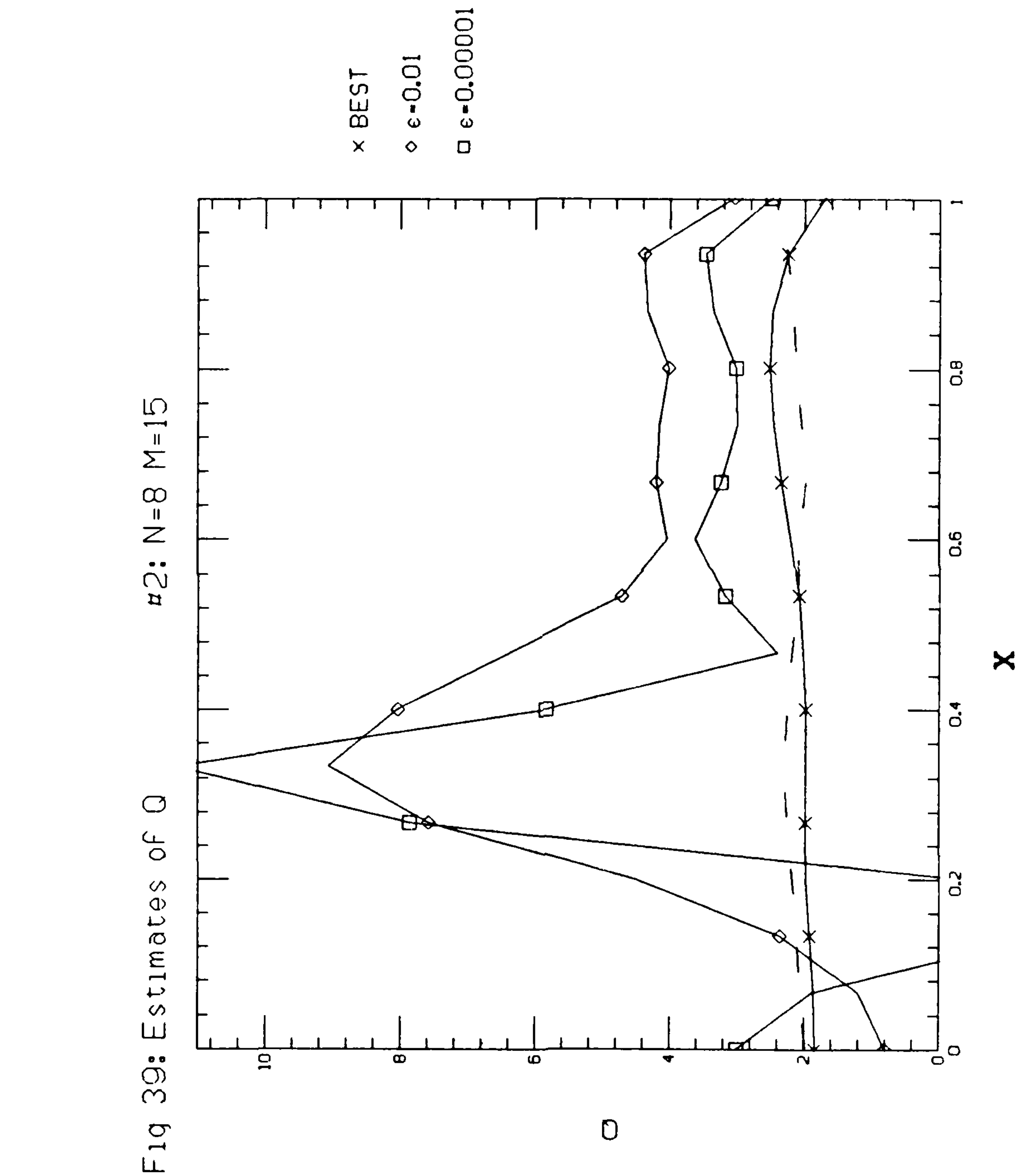
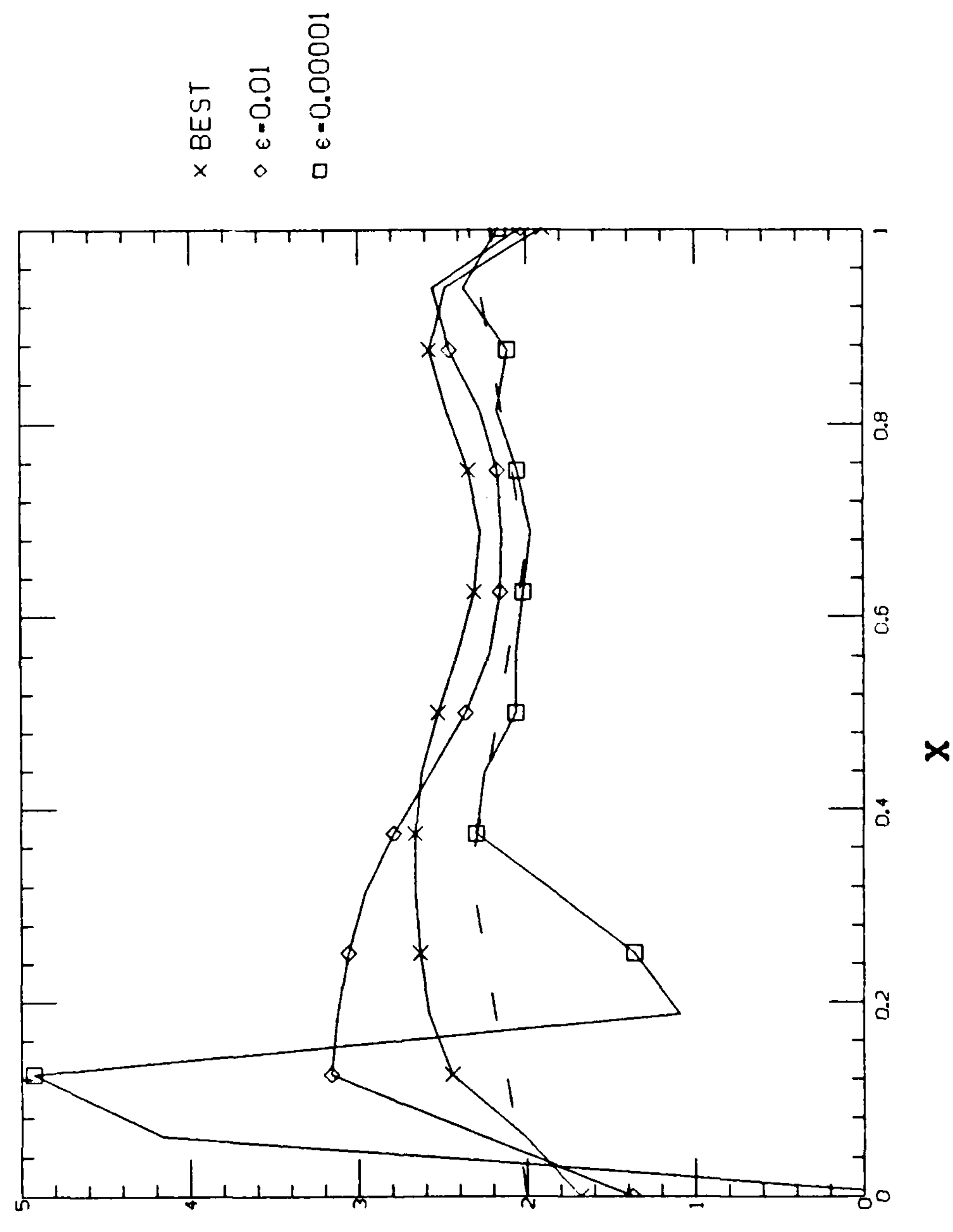


Fig 40: Estimates of Q   #2: N=16 M=16



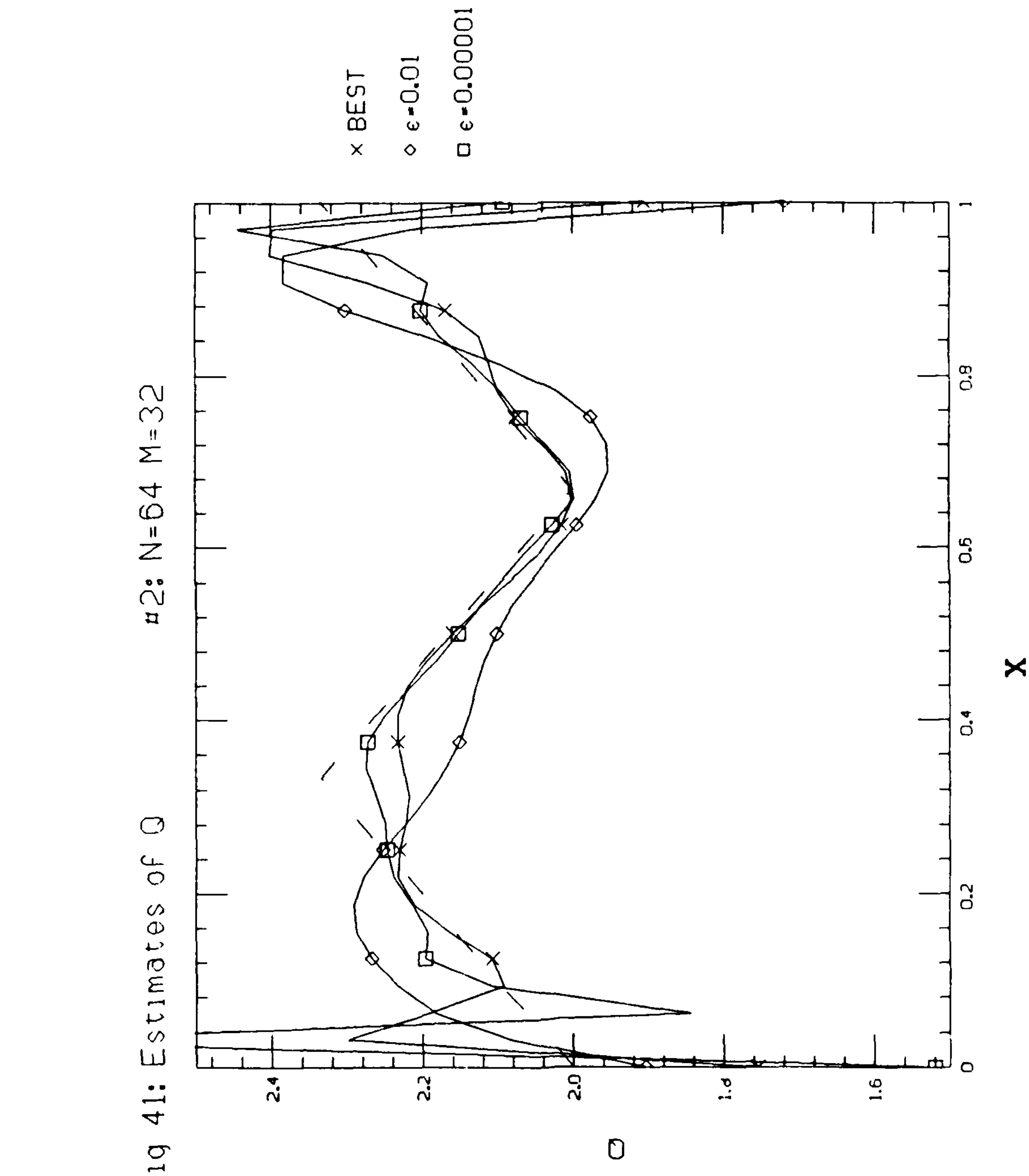


Fig 41: Estimates of Q

Conclusions

The work in this paper demonstrates severe problems in some instances with using an unconstrained algorithm to estimate the parameter q. When modified, either by regularizing the problem using Tikhonov regularization or by constraining the estimate set ~~as in this paper~~, the algorithm does give good estimates.

Unlike the unconstrained algorithm, both the Tikhonov and constrained algorithms are stable with respect to increasing M while holding N fixed. However as N is increased the estimates from the Tikhonov algorithm do not improve as much as do those of the constrained algorithm. The Tikhonov estimates are biased by the regularization of the cost functional, and never show all the detail of q when q has significant variation.

Both the constrained and Tikhonov estimation algorithms are stable with respect to systematic errors in the input data, while, except when N is large, the unconstrained algorithm fails to give good results on even the exact data.

For both the Tikhonov and constrained algorithms there are parameters which affect the algorithm's performance. For the constrained algorithm suitable constraints must be found while for the Tikhonov algorithm suitable values of $\overset{a}{\alpha}$ and $\overset{b}{\beta}$ must be found. The constrained algorithm has the advantage that the constraints used here, i.e. limits on the slope of q, have an obvious meaning, and so may well be known in advance. In the Tikhonov algorithm β and α have no obvious meaning. They must be suggested by looking at the change in the estimate behavior as β and α change, and perhaps using some apriori knowledge about the shape of q to choose values of β and α that give an estimate that is neither too flat, nor too oscillatory.

Appendix A The Reduced Gradient Optimization Algorithm

For each M, q^M is represented as a linear combination of linear splines. The set of such functions is a subspace of C[0,1] which is isomorphic to $\mathbf{R}^{M+1}$. The bounded derivative condition in C[0,1] becomes a condition on the difference between successive coefficients in $\mathbf{R}^{M+1}$, i.e. $|\rho_{j+1}-\rho_j|$ must be bounded. The condition that the function be bounded becomes a condition on the values of the individual coefficients, i.e. $|\rho_j|$ must be bounded.

Thus the minimization is over a subset of $\mathbf{R}^{M+1}$, this subset being defined by a set of constraint equations. To find the minimum subject to these constraints the reduced gradient algorithm [6] was used.

In the reduced gradient algorithm the coefficients of q^M are transformed by a linear transformation to give coefficients with respect to a new (in general, nonorthogonal) coordinate system in which the constraints are rectangular. The minimum is then found by a modified gradient algorithm. At each iteration there is a linear search, the direction of which is given by the gradient of $J(q^M)$, with respect to the new coordinate system, modified by setting to zero any components which would cause the linear search to violate a constraint.

The reduced gradient algorithm was also used when there were no constraints, so the results could be compared with the results for the constrained case. Although there were no constraints the coefficients were still transformed using the same linear transformation as for the constrained case. However in this case the linear searches were in the direction of the unmodified gradient vector, there being no constraints to violate.

In both the constrained and unconstrained cases a diagonal search direction, i.e. $q^M_i - q^M_{i-1}$, the direction given by the difference between the two previous estimates of q, was used whenever

(a) the dot product of the two previous improvement vectors was negative, i.e.

$$(q^M_i - q^M_{i-1})\cdot(q^M_{i-1} - q^M_{i-2}) < 0,$$

and (b) neither q^M_i nor q^M_{i-1} were the result of a diagonal step.

Appendix B Finding u^N as a function of q^M

In this section the tensor summation notation is used, i.e. when an index appears as both a subscript and a superscript in a product term it represents the sum of the product over the repeated index.

Substituting (5) and (7) into (3) we obtain

$$(23) \qquad < \sum_{j=1}^{M+1} \rho_j \psi_j \sum_{i=1}^{N-1} \sigma_i D\phi_i, D\phi_k > + <f, \phi_k> = 0, \qquad k=1,\ldots,N\text{-}1$$

or

$$\sum_{j=1}^{M+1} \sum_{i=1}^{N-1} \rho_j \sigma_i < \psi_j D\phi_i, D\phi_k > + F_k = 0, \qquad k=1,\ldots,N-1$$

which is equivalent to

$$\rho_j M_k^{ij} \sigma_i + F_k = 0$$

where

$$(24) \qquad F_k = <f, \phi_k>$$

$$(25) \qquad M_k^{ij} = <\psi_j D\phi_i, \phi_k> = \int_0^1 \psi_j D\phi_i \phi_k dx$$

$$= \begin{cases} -\int_{(p-1)/N}^{p/N} \psi_j dx \equiv -I_p^j, & i=k+1=p \text{ or } k=i+1=p \\ \int_{(i-1)/N}^{(i+1)/N} \psi_j dx = I_i^j + I_{i+1}^j, & i=k \\ 0 & \text{otherwise.} \end{cases}$$

Thus M_k^{ij} are symmetric positive definite tridiagonal matrices for each j. So $A_k^i = \rho_j M_k^{ij}$ is also a symmetric tridiagonal matrix, and is positive definite if $\rho_j > 0$ for all j.

Thus

$$A_k^i \sigma_i + F_k = 0, \qquad k = 1, \ldots, N-1 \tag{26}$$

has a unique solution.

The matrix A_k^i, being an n x n symmetric and tridiagonal matrix, has an LDL^t–decomposition where L is lower triangular with one's on the diagonal and one non-zero lower diagonal and D is diagonal ([4] with $DL^t = U$).

i.e.

$$L_{ij} \begin{cases} = 1, & i=j,\ 1 \le i \le n \\ = \mu_i, & j=i-1,\ 2 \le i \le n \\ = 0, & \text{otherwise.} \end{cases} \tag{27}$$

$$D_{ij} \begin{cases} = \gamma_i, & i=j,\ 1 \le i \le n \\ = 0, & \text{otherwise.} \end{cases} \tag{28}$$

Because of the form of M_k^{ij}, A_k^i has the following special form :

$$A_k^i \begin{cases} = X_i + X_{i+1}, & i=k \\ = -X_i, & i=k+1 \\ = -X_k, & i=k-1 \end{cases} \tag{29}$$

where

$$X_i = \rho_j I_i^j \tag{30}$$

With this form γ_i and μ_i are given by :

$$\begin{aligned} \gamma_1 &= X_1 + X_2 \\ \gamma_i &= C(i) / C(i-1) & 2 \le i \le n \\ \mu_i &= -X_i/\gamma_{i-1} & 2 \le i \le n \end{aligned} \tag{31}$$

where C(r) is the sum of the r+1 products of possible combinations of r I_i^j 's out of the first r+1 I_i^j 's (i = 1 . . r+1).

e.g. $C(2) = I_1^j I_2^j + I_1^j I_3^j + I_2^j I_3^j$

Proof

Substituting the elements of A_k^i into the equations for the LU decomposition of a general tri-diagonal matrix [4],

the values of γ_i and μ_i are given by :

$$(32) \qquad \gamma_1 = X_1 + X_2$$

$$(33) \qquad \gamma_i = X_i + X_{i+1} - X_i^2/\gamma_{i-1} \qquad i = 2 \,..\, n$$

$$(34) \qquad \mu_i = X_i/\gamma_{i-1}$$

Assume (31) is true for some i, then using (33)

$$(35) \qquad \gamma_{i+1} = (X_{i+1} + X_{i+2}) - (X_{i+1})^2 \, C(i-1)/C(i)$$

$$= C(i+1)/C(i)$$

so (31) is also true for i+1. (31) is true for i=2 so by induction it is true for $2 \le i \le n$.

The value of C(i) is easily calculated from C(i−1) so the calculation of γ_i and μ_i and hence of the LDL^t– decomposition can be done very quickly and easily. However C(i) grows (or declines) exponentially with i. In a numerical implementation this problem can be solved by calculating $C(i)/\mu^i$ where μ is the geometric mean of the I_i. In the programs used here μ was approximated by taking the geometric mean of I_1, $I_{INT(N/2)}$, and I_N.

Given the LDL^t–decomposition of A_k^i equation (26) can be solved to give u^N as a function of q^M by simple back substitution.

Appendix C Implementation Details

The evaluation of integrals by Simpson's Rule

The quantities $J(q^M)$, $F_i = <f,\phi_i>$ and $\| q^M \|_2$ were evaluated using Simpson's rule with N1 subdivisions of the interval [0,1] for the first two and N2 subdivsions for the last one. The quantity $\| q^M \|_\infty$ was evaluated by sampling the difference at the same N2 points used in the Simpson's rule evaluation of the L_2 norm.

N1 and N2 were always at least twice N and M respectively. Only the value of N1 affects the algorithm. The accuracy of the integration will be most sensitve to changes in N1 when one of N or M is not a multiple of the other. In this case the product $f \cdot \phi_i$ has a discontinuous derivative at the points i/N and j/M, and the j/M points are not, in general, included in the sum for the Simpson's rule evaluation of the integral of the product $f \cdot \phi_i$.

The value of N2 does not affect the convergence of the algorithm, it affects only the evaluation of the L_2 and L_∞ norms, which were used for informational purposes and for determining convergence but not in the calculation of the successive steps of the optimization.

The implementation of the algorithms

The algorithms were implemented in FORTRAN on an IBM3081. Double precision arithmetic was used throughout.

References

1. H.T. Banks, P.L. Daniel, and E.S. Armstrong, A spline based parameter and state estimation technique for static models of elastic surfaces, ICASE Rep #83-25, NASA Langley Res. Ctr, Hampton, VA, June 1983.

2. H.T. Banks, P.K. Lamm, and E.S. Armstrong, Spline-based distributed system identification with application to large space antennas, AIAA J. Guidance, Control and Dynamics, May-June, 1986, to appear.

3. H.T. Banks, On a variational approach to some parameter estimation problems, ICASE Rep No 85-32, NASA Langley Res. Ctr, Hampton, VA, June 1985.

4. G. Dahlquist and A. Bjorck, *Numerical Methods*, translated from Numeriska metoder. Prentice-Hall,Inc.,Engllewood Cliffs,N.J., (1974), 166-167.

5. C. Kravaris and J.H. Seinfeld, Identification of parameters in distributed parameter systems by regularization, SIAM J. Control and Optimization *22*, (1985), 217-241.

6. P. Wolfe, Methods of nonlinear programming, in *Nonlinear Programming*, (J. Abadie,ed.), North-Holland Publishing Company, Amsterdam, (1967), 97-131.

7. Y.S. Yoon and W.W-G. Yeh, Parameter identification in an inhomogeneous medium with the finite-element method, Soc. Pet. Engr. J, *16*, (1976), 217-226

END

1-87

DTIC